Service Embedding in IoT Networks

Haider Qays Saeed Al-Shammari

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds
School of Electronic and Electrical Engineering

June 2019

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Chapter 3 based on the work:

> Haider Al-Shammari, Taisir Elgorashi, Jaafar Elmirghani, "Service virtualisation in IoT: A survey", to be submitted to IEEE communication survey and tutorials.

Prof. Elmirghani, the supervisor, suggested the survey on the service virtualisation in IoT networks. The co-supervisor, Dr El-Gorashi, and the supervisor worked with the student on the research and articles to be surveyed in paper preparation.

Chapter 4 based on the work:

> Haider Al-Shammari, Ahmed Lawey, Taisir Elgorashi, Jaafar Elmirghani, "Energy-efficient service embedding in IoT networks"; WOCC 2018.

Prof. Elmirghani, the supervisor, suggested the virtualization in IoT networks. The co-supervisor, Dr. El-Gorashi, reviewed the MILP model and the proposed architecture. The former co-supervisor Dr Lawey suggested the deployment of virtualisation in IoT networks. The PhD student developed MILP model, obtained and analysed results and wrote the paper.

Chapter 5 based on the work:

> Haider Al-Shammari, Ahmed Lawey, Taisir Elgorashi, Jaafar Elmirghani, "Energy-efficient service embedding in IoT over PON network", ICTON July 2019.

And

> H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Service embedding in IoT networks," submitted to IEEE Access.

Prof. Elmirghani, the supervisor, suggested the expansion of the work. The co-supervisor, Dr. El-Gorashi, suggested the deployment of PON networks in the proposed architecture. The former co-supervisor Dr Lawey suggested the deployment of virtualisation in IoT networks. The PhD student developed MILP model, obtained and analysed results and wrote the paper.

Chapter 6 is based on work:

> H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Resilient service embedding in IoT networks" submitted to IEEE Access.

Prof. Elmirghani, the supervisor, suggested the traffic resilience study and reviewed the MILP model and discussed the results. The co-supervisor, Dr El-Gorashi, worked with the student on the MILP model development, results analyses and paper preparation. The former co-supervisor Dr Lawey suggested the deployment of virtualisation in IoT networks and the study of resilience. The PhD student developed the model, obtained and analysed the results, and wrote paper.

© 2019 The University of Leeds and Haider Qays Saeed Al-Shammari

Acknowledgements

My parents, the most important people through all stages of my life, I dedicate this thesis to them.

My deep gratitude to my supervisor, Prof Jaafar Elmirghani, for his continuous support and insightful discussions about the research during my study. I learned countless lessons from his knowledge and experiences, and I am very proud to have had him as my supervisor.

I am also hugely appreciative to Dr Taisir El-Gorashi, my co-supervisor, for her valuable and supportive advice and rewarding discussions.

I would like to acknowledge my friends Dr Ahmed Lawey, Dr Ahmed Al-Qizweeni and Mohammed Hadi for their worthwhile encouragement and assistance.

I would also like to thank the Higher Committee for Education Development in Iraq, Iraqi Prime Minister Office, for the sponsorship and support extended to me.

Last but not the least, I would like to express my sincere gratitude to the wonderful academic staff and research community of University of Leeds for their advice and collaboration.

Abstract

The Internet of Things (IoT), also called the Internet of Everything, is a new technology that has realised the paradigm of a global network of things capable of interacting with each other. IoT nodes can generate enormous amounts of data, perform certain analyses, and make decisions to provide efficient and smart services. These data are used to discover and resolve various issues and to provide value-added services to the users. The data analysis and decision making may be embedded into IoT devices to ensure that the decision-making takes place at the data source (i.e. the sensor node).

The IoT infrastructure is composed of numerous heterogeneous devices used and deployed by various applications and services. The recent IoT architectures are designed as service- or event-driven paradigms, and there is no unified IoT architecture to correlate and integrate the data from physical nodes and services. The proposed IoT solutions use private techniques and cause several problems related to technology compatibility, information sharing, service management, and network upgrading. All these obstructions are impeding the development of IoT. In order to integrate various resources and services into a comprehensive system, these is a need for an efficient architecture that hides such heterogeneity from higher-level applications, provides interoperability for information exchange with other IoT devices, and considers different service scenarios, application-based demands, and recent technologies.

Energy efficiency is considered a key enhancing factor in IoT where the sensing, processing, and communication of a huge number of IoT devices consume substantial amounts of energy. Energy consumption is considered to be a sustainability issue with respect to IoT devices, as these devices are powered by low-power sources or batteries, which impede the continuous operation of IoT systems. Recent proposals have increased the energy efficiency of processors and networks through a further development of energy-efficient paradigms.

In this thesis, these motivations were considered to propose and develop a framework for service embedding in IoT networks to enhance the network performance and reduce both the cost of devices and the power consumption.

Furthermore, a service-oriented architecture (SOA) was designed as viable middleware between a user's applications and an IoT physical layer, and interoperability between heterogeneous IoT devices was achieved. This SOA enabled the abstraction of IoT device functions that could then be translated into basic services, which in turn could be composed of complex services and exploited to the upper application layer.

Energy efficiency was also considered a key enhancing factor in the proposed framework, as processing and communication power consumption is a sustainability issue in IoT systems. The objective of minimising the power consumption resulted in a framework that selected only low-power-consumption nodes and routes. The constraints introduced while minimising power consumption affected network-related issues such as traffic latency because of mutually dependent factors including the traffic volume and the routes selected to consolidate the traffic and hence save power. The proposed framework also optimised the selection of the routes for the traffic between the source and the destination to enhance the optimality of power saving and latency minimisation.

Furthermore, we extended the proposed framework to include the fog and the cloud as on-demand access processing resources, as real-time services require a high processing speed and considerable data storage. We investigated the impact of the processing latency and the coexistence constraints of the processing and network power consumption. The cloud and the fog provided an ideal processing solution for IoT devices and services.

In IoT networks, a node can cause temporary outages. A smart building consists of interconnected sensors, controllers, and actuator devices. If a sensor of a monitoring and control system fails, the controller may receive an incorrect signal, which may result in the failure of the entire system. Hence, node and traffic resilience were investigated in this thesis. Furthermore, recent resilience schemes were evaluated in terms of their power consumption and mean traffic latency. As a result, in this thesis, a novel traffic resilience technique was developed to enhance network performance and reduce network power consumption.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The ubiquity and intelligence of the Internet of Things (IoT) in addition to the sensing/actuating abilities and wireless connectivity are becoming the cornerstone in the design and development of smart domains, such as smart buildings, smart transportation and smart cities. These smart domains are seamlessly and efficiently enhancing our daily lives. It is predicted that the number of IoT nodes will reach 50 billion by the year 2020 [5]. This growth comes with challenges in terms of efficient utilisation of resources, power consumption, traffic congestion and security.

The decentralised and heterogeneous properties of IoT devices capable of providing multiple functions require an efficient architecture that hides such heterogeneity from higher-level applications and provides interoperability for information exchange with other IoT devices [8]. The Service Oriented Architecture (SOA) is a new paradigm that focuses on the services to be supported instead of focusing on the network hardware components. It is therefore considered a viable middleware between users' applications and the IoT physical layer and can support the interoperability between those heterogeneous IoT devices [9]. SOA enables the abstraction of IoT devices so upper application layers can embed virtual nodes of certain functions into IoT nodes with physical resources that support these functions to compose complex services. The abstraction of IoT resources allows the virtualisation of multiple of these services over the same IoT resources [1], [2].

In this thesis, we developed a framework for embedding services requested by the application layer into the substrate network wirelessly connected IoT nodes in smart buildings and smart cities domains. We focused on optimising the service embedding in IoT networks to improve energy efficiency and reduce the traffic latency. We formulated the IoT virtualisation problem using Mixed Integer Linear Programming (MILP). We also studied service embedding in IoT networks integrated with cloud and fog computing to meet the increasing needs of IoT applications that cannot be met by the limited processing and storage resources of

the IoT devices. Resilient IoT virtualisation is also considered together with its impact on power consumption and traffic mean latency.

1.1 Research Objectives

The research objectives of this thesis can be summarised as follows:

1- To develop a framework for service embedding in IoT networks in a building setting and to study minimising the power consumption of the IoT nodes and the network and minimising the latency experienced by traffic.
2- To study the integration of the IoT layer with the fog and the cloud to support services of high processing demands in a smart city setting.
3- To study the embedding of delay-sensitive services in the IoT-fog-cloud integrated architecture.
4- To investigate the impact of improved node and network resilience of service embedding in IoT networks in terms of power consumption and queuing latency.

1.2 Original Contributions

The main contributions of this thesis as the following:

1- Mathematically modelled the problem of service embedding in IoT networks using MILP and evaluated the power consumption and queuing delay of service embedding considering a building setting.
2- Developed a heuristic to verify the MILP model results and to support real time service embedding in IoT networks.
3- Studied the impact of embedding in different geographical zones and studied geographical coexistence of virtual nodes paying attention to power consumption and queuing delay of IoT service embedding in the building setting.

4- Proposed an IoT-Fog-Cloud integrated architecture and modelled energy efficient service embedding in this architecture using MILP considering a smart city setting.

5- Developed a MILP to optimise the embedding of resilient service in IoT networks and evaluated the power consumption and queuing delay resulting from different levels of node and network resilience.

1.3 Related Publications

The original contributions in this thesis are supported by the following publications:

1- H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy efficient service embedding in IoT networks," in Wireless and Optical Communication Conference (WOCC), 2018 27th, 2018: IEEE, pp. 1-5.

2- H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy efficient service embedding in IoT over PON," accepted in 21th International Conference of Transparent Optical Network (ICTON), July 2019.

3- H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Resilient service embedding in IoT networks," submitted to IEEE Access.

4- H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Service embedding in IoT networks," submitted to IEEE Access.

5- Haider Al-Shammari, Taisir Elgorashi, Jaafar Elmirghani, "Service virtualisation in IoT: A survey", to be submitted to IEEE communication survey and tutorials.

1.4 Thesis Structure

Following the introduction in Chapter 1, the rest of the thesis is organised as follows:

- Chapter 2: This chapter provides a literature review of the evolution of the Internet of Things and the architectures proposed in the literature. It also explains the IoT elements and applications and recent challenges and solutions.
- Chapter 3: This chapter presents a survey of service virtualisation in IoT. It reviews recent architectures for virtualization in IoT and summarises their advantages. The chapter reviews the challenges and open issues related to the service virtualisation in IoT.
- Chapter 4: This chapter introduces a novel framework for embedding services into an IoT network in smart buildings' settings. A MILP model for minimising power consumption and traffic mean latency is developed. A heuristic is also developed to verify the model and provide real time solutions for the embedding problem.
- Chapter 5: This chapter presents an extension of the IoT architecture presented in Chapter 4 where the IoT network is integrated with the cloud and fog to meet the growing demands of IoT applications in smart cities. The model also investigates the impact of processing splitting, processing latency, and coexistence constraints on the processing and network power consumption.
- Chapter 6: This chapter introduces an extension to the same setting of Chapter 4 of resilient service embedding. We present a framework for energy efficient- low latency resilient service embedding in smart buildings. The

chapter investigates the different levels of node and traffic resilience and evaluates them in terms of power consumption and traffic mean latency.

- Chapter 7: In this chapter, the thesis conclusions are drawn, and the major contributions of this work are summarised. Future directions are also proposed.

Chapter 2 The Internet of Things

2.1 Introduction

The Internet of Things (IoT) is an innovative paradigm that exploits advanced wireless communications to interconnect different objects such as sensors, actuators, mobile phones, cars, etc. These objects communicate with each other to implement specific services [3]. Historically, the trend “Internet of Things” has been initiated by MIT Auto-ID centre in 1999 [4], [5]. The term “Internet” refers to a global network that interconnects all things, surrounding humans, using advanced communication technologies while the term “Things” refers to the physical object that has the ability to interact with each other and cooperate with their neighbours to perform common goals [3].

After 2005, IoT became one of the most important concepts globally and has driven many research areas according to the International Telecommunication Union (ITU) [4], [6], [7]. The IoT is considered an integrated part of the future Internet due to the growth in smart applications. Recent reports indicate that the expected number of connected IoT devices will reach approximately 50 billion by the year 2020 [8], [9]. The great number of interconnected devices increases the demands for planning and consideration of important factors such as addressing, mobility, reliability, coverage, link capacity, energy efficiency, and device cost [10], [11], [12], [13].

2.2 Evolution of the IoT

Historically, the phrase "Internet of Things" or "IoT" was introduced in 1999 by Kevin Ashton the founder of the original MIT Auto-ID Centre [14], [15]. The concept of "Internet of Things" was officially announced in 2005 when the International Telecommunications Union published the first report on the “Internet of Things” [8].

Technically, the IoT has is based on a range of different emerging technologies as explained below:

2.2.1 Radio Frequency Identification (RFID)

RFID is a non-contact communication technology for identification applications and services in which the reader receives data from the tagged objects through electromagnetic fields or static electronic coupling [16], [17]. RFID depends on two main parts:

- An embedded tag attached to a person, product, materials or anything. The tag contains identification information.
- The Reader, which produces a wireless signal at a specific frequency and range of power.

When the tag passes though or is near the reader, the tag receives the interrogator signals and sends an authentication response to the reader. There are also other tag and reader classifications according to the required level of security, cost, and size. An important type of RFID is Near Field Communication (NFC) based tags, which can provide contactless communication for cards and mobile phones [10], [18].

2.2.2 Wireless Sensors Network (WSN)

This wireless network technology interconnects a huge number of distributed, possibly autonomous, sensors to monitor and control systems or the environment. The WSN combines advanced technologies such as information and communication technologies, integrated sensing and computation, [19]. WSNs are widely used because of their low-cost, energy efficiency, wide distribution, and self-organization properties [20]. WSNs play an important role in the evolution of the future Internet, providing sensing and actuation capabilities required by future applications [21]. WSNs play an important role in the IoT perception layer, where the WSN is responsible for sensing, monitoring, and tracking the status of the devices and the environments. In addition to sending the data to the control unit through the network [22].

2.2.3 Machine-to-Machine (M2M) Communication

This is a communication mode that provides a connection between multiple-systems and remote devices that directly exchange information. It also refers to communication that provides data to applications running in devices [23]. IoT uses

different communication modes such as Thing-to-Human communication, Human-to-Thing communication, and Thing-to-Thing communication [24]. The idea of M2M communication has major advantages and enables the construction of comprehensive connections between machines and other entities [25].

2.2.4 Cloud computing

The cloud can provide computation and data storage services to the IoT layer and can provide other services according to network scale and application demand [18]. The integration of cloud computing and IoT has resulted in a range of new services witnessed recently such as:

- Infrastructure as a Service (IaaS) [26].
- Platform as a Service (PaaS)[27].
- Software as a Service (SaaS)[28].

The researchers and stakeholders have increased their efforts directed towards the integration of cloud computing with IoT, to tackle the challenges associated with the huge data storage needed and the computing and processing required [29]. Due to the low cost, small size and low power consumption of IoT devices together with their limited data processing, storage, and traffic handling capabilities, IoT devices, are often supported by a coordinator that boosts their capabilities and provides additional functions. Cloud and Fog play important roles as main contributors to coordinate and provide data storage, resource management, service creation, service management, service discovery, and power management. The integration of IoT with Cloud leads to a new paradigm that can result in the success of IoT in terms of service provisioning, high-performance, reliability, ubiquity, and scalability. The cloud features can be provided with high elasticity and on-demand for efficient and scalable service provisioning [30], [31], [32], [33], [34], [35], [36], [37].

2.3 IoT Architecture

There are a number of research efforts devoted to developing IoT prototype architectures [38], [39], [40], [41], [42]. In this section, we review two of the popular IoT architectures:

2.3.1 Three-layer architecture

This architecture is considered as one of the most popular architectures. It is divided into the following three layers as shown in Figure 2-1 [8], [43], [44]:

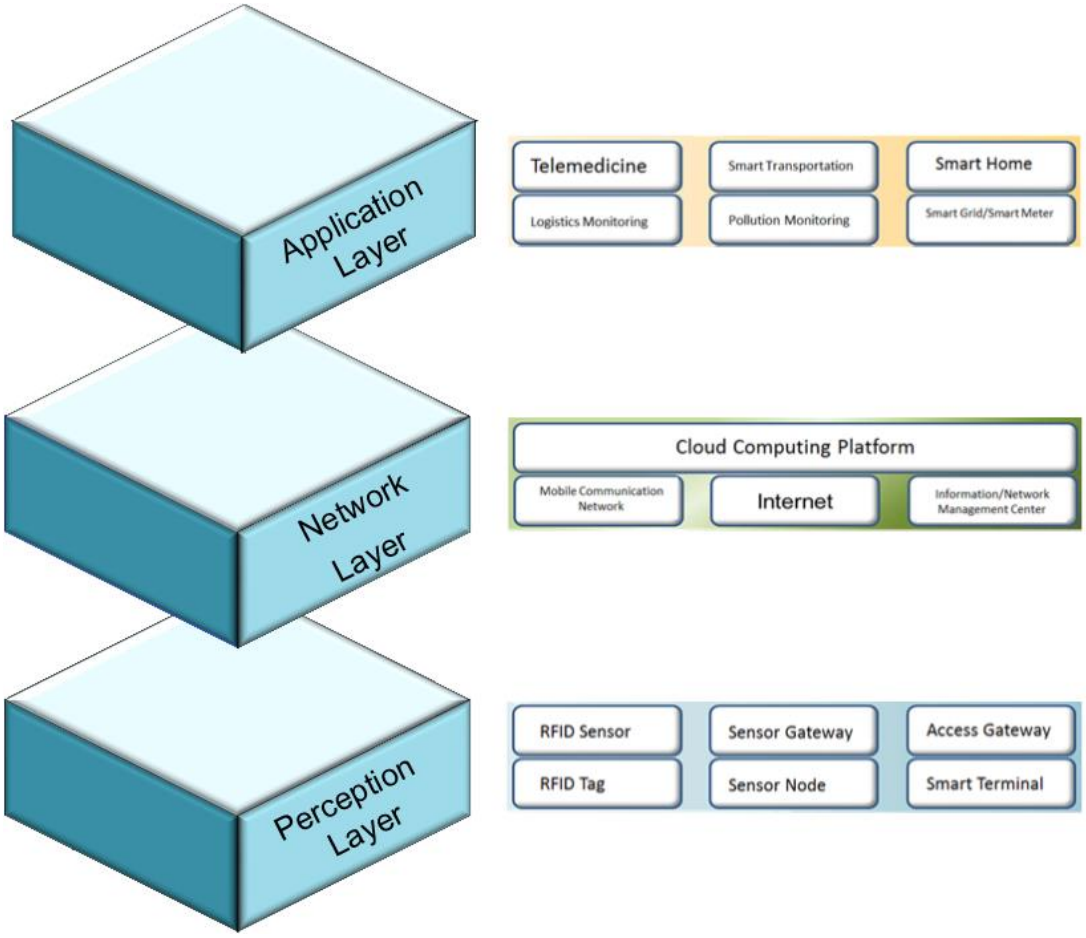


Figure 2-1: Three-layer IoT architecture

- Application Layer: This layer is the highest layer. It manages different services and applications and provides information and functional access to relevant sensors with corresponding applications. It receives the transmitted data from the network layer and uses the data to provide the required services or applications [41]. Several applications are widely known in this layer such as smart city applications, smart grid, smart transportation, etc.
- Network Layer: This layer is the most important layer in IoT architecture. It builds the network topology and routes the data and information produced by the IoT nodes. The network layer is responsible

for sending and receiving data between the applications and heterogeneous devices through interfaces. It uses a range of communication technologies and protocols.

- Perception layer: This layer is known as the sensors layer or physical layer also. It is responsible for the interaction of physical devices such as sensors, actuators, etc. It collects and/or processes the state information of these devices and sends the information to the upper layers.

2.3.1 Four Layer architecture

The IoT architecture is usually characterised by the mentioned three layers paradigm in which the state information is collected by the perception layer and transmitted to the network layer which manages the connection and provides data to the application layer, where the latter provides the services [39]. The traditional three-layer architecture can be expanded by adding a new layer, namely the middleware layer (also known as a service layer or the interface layer). This results in the four-layer IoT architecture made up of the perception layer, network layer, middleware layer, and application layer. The new middleware layer can be defined as a software layer between the application and network layers. The middleware is essentially responsible for service programming. It provides an abstraction between IoT technologies and applications [45], [46], [47]. The middleware hides the details of different technologies and provides compatibility between applications and infrastructures [47]. The middleware allows the devices and applications to exchange information and share resources with each other. The middleware layer has many advantages [41], it:

- Enables the co-existence of various applications seamlessly.
- Enables various operating systems, platforms, and protocols.
- Supports distributed computing and the interaction of services among heterogeneous networks, devices, and applications.
- Solves incompatibility between standard interfaces, providing portability and standard protocols to enable interoperability and standardisation.
- Enhances high-level interface stability for applications. The stable interfaces allow the applications to deploy the hardware and operating system independently.

These advantages make middleware an appropriate paradigm for the IoT, because of the huge number of decentralised and heterogeneous devices and networks that exist. Thus, there is a high need for integration and continuous updating to serve different applications. The IoT consists of a huge number of ubiquitous devices. These devices have many embedded components including sensors, actuators, and serve multiple applications using the architecture adopted [48]. The IoT devices have standards that enable them to collect specified data for identified things and processes, transmit the information, and perform actions without direct human intrusion. This process realises the interconnection paradigm between the physical world and cyber world [45], [46], [49], [50].

2.4 IoT Elements

The IoT has the following key elements as shown in Figure 2-2 [51]:

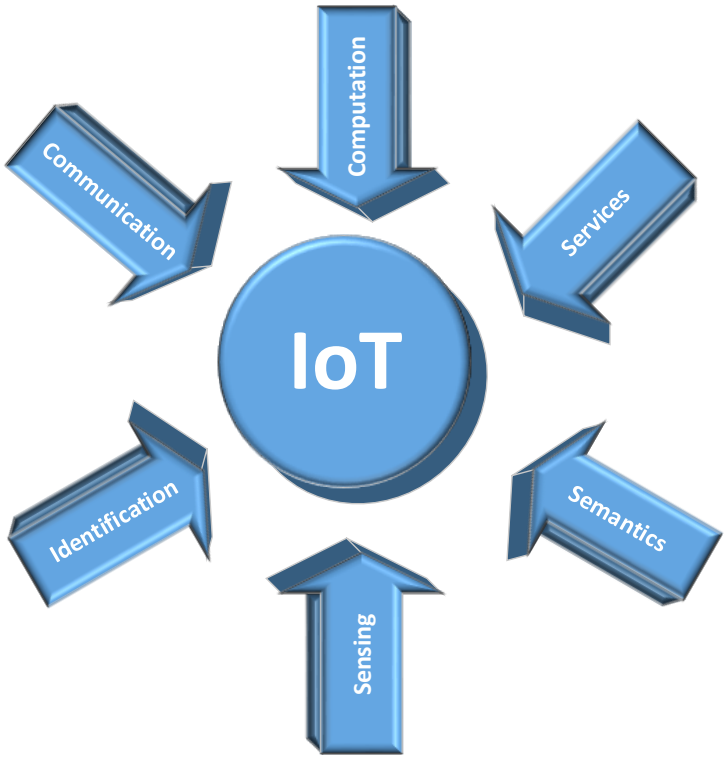


Figure 2-2: The key elements of the IoT

- Identification

  Identification is crucial for the IoT to name and match services with their demand [52]. Furthermore, in addressing the IoT objects it is important to differentiate between the object ID and its address [53]. The object ID refers to its name such as “T1” for a particular temperature sensor, while the object’s address refers to its address within a communications network [54].

- Sensing / Actuation:

  The main physical components in IoT nodes are sensors and actuators. The sensors can be defined as detectors for physical entities that read and collect information about specified things. Usually but not always, the sensors have low cost, low power consumption and limited processing ability, and have interfaces to communicate through specified communication channels. There are many types of sensors dedicated to sensing the physical environment and thus measure quantities such as temperature, acceleration, vibration, light, electromagnetic properties, humidity, and the positions of physical things. According to the sensed data, the sensors can be classified into two types as follows [53], [55], [56]:

  1- ID-Based sensors: This type of sensor detects things with identifiers, e.g. FRID tags.

  2- nID-Based sensors: This type of sensor measure physical attributes, e.g. temperature sensors.

- Communication:

  Many communications Technologies support IoT devices to deliver their smart services. Examples of communication protocols used for the IoT are WiFi [31], Bluetooth [57], IEEE 802.15.4, and cellular Long-Term Evolution (LTE). These communication technologies have varying capabilities in terms of the data rates they can support, the distances they can span and the power they consume. [58], [59], [60].

- Computation:

  Computation is needed in general to process the data collected to enhance the performance of a physical system. The computation types can be classified as follows [30], [34], [61], [62]:

  1- Centralized Computation mode:

  This mode operates with the controller as an independent functional module with satisfactory computation and communication abilities. The sensors send the collected data to the controller. The controller applies control algorithms and assigns task orders to the actuators. The actuators receive the control orders and perform the required actions.

  2- Distributed Computation mode:

  This mode operates with the actuator and controller (processor) on the same device. The sensors collect data and send it to the corresponding controller/actuators. The controller applies control algorithms and performs the action required as an actuator. In other words, the controller and actuators are integrated in the same entity, and the sensed data are collected by actuators with control orders to act.

- Services

  In general, the services provided by IoT have been deployed in many aspects of our life [32]. With the widespread use of wireless communication, the range of IoT applications has become much more comprehensive i.e. Building services, Power and cooling services, Safety services, Industrial automation services ...etc. [34], [63], [64], [65], [66], [67].

- Semantics

  In the IoT, semantics can be defined as the ability to construct knowledge by using different techniques to deploy smart services. The knowledge construction includes discovery and use of the resources, and modelling and analysis of the information, then making the logical decision to deploy smart services [68].

2.5 IoT Applications

There is a huge number of applications related to IoT and it is thus difficult to summarise all of them, consequently, this section summarises the recent IoT applications according to their domain [69]. In this section, we review a number of these applications according to their domain as follows:

2.5.1 Personal appliances

There are many applications that address personal requirements. The sensors are used in this case only by the person who directly owns the devices. Application in this area has been proposed for personal healthcare, support, navigation, and to analyse the person's environment using surrounding sensors. Recently, the smartphones have been used as communication gateways to provide several connections including Wi-Fi and Bluetooth to connect sensors measuring environmental parameters [70]. The applications available use the operating systems of the smartphones. They sense and act on several parameters. These applications can be centralised in the cloud or decentralised in local devices [7]. With the development of IoT personal products, many industries provide innovative efficient low power solutions for wearables appliances, such as entertainment and fitness devices, smartwatches, location tracking devices. [10].

2.5.2 Home Automation

The deployment of IoT in the form of connected home devices can create a home automation system, which allows the user to monitor, operate and control their homes using simple applications. The increasing number of home devices supported by IoT provides a global framework that can be used to develop home automation applications, enhance security level, and provide more new applications. Currently, there are many home automation applications, varying from enhancing security to reducing energy and maintenance costs. IoT operators can provide a wide range of IoT technologies for the control and monitoring of smart homes and buildings. The functions can include such access control, energy saving, and security [25], [71], [72], [73], [74], [75], [76].

2.5.3 Environment Applications

There is a huge number of sensors embedded in the environment connected by IoT for enterprise applications. These sensors collect data from different zones, over some time to be analysed by specialists or released socially. The most common applications related to environmental monitoring, collect all information and forward it to data centres. Usually, these types of sensors are specialised and contain integral applications, factory configured to ensure security, functionality, and control. Wireless communication is driving environment applications by providing coverage and flexibility. There are many well-known environment applications; with additional applications planned in the future. There are however applications that have drawn attention, namely smart environment applications [17]. A smart environment can be realised by embedding sub-systems that can sense /act on one or more parts of the environment. These sub-systems are creating an urban environment, which exploits the benefits of IoT and realize it on the ground in the form of smart cities, smart agriculture and related [77], [78], [79].

2.5.4 Medical and healthcare systems

The IoT can be used to develop health monitoring systems to sense, monitor, analyse, and interact with the patients and provide communication to doctors. The IoT devices used vary from simple sensors to complex advanced devices [80]. These specific sensors can be installed in the area surrounding elder people or patients anywhere. The devices can include wearable small devices such as blood pressure monitors, heart rate monitors to advanced devices like pacemakers [18]. These devices will ensure appropriate treatment is being directed to assist the patients anytime and anywhere. Other medical applications involve using sensors to support a healthy living style. The IoT devices used include wearable heart monitors, calorie counters, training observers...etc. [35], [81]. There are many additional medical and health applications that make use of IoT platforms to support and treat chronic patients and provide emergency support [35], [36], [80], [82], [83], [84].

2.5.5 Manufacturing and Industry

In the manufacturing and industry sectors, IoT applications provide the ability to monitor and control the product lines and devices. These applications are based on

sensing technologies to collect the information and analyse the status to enhance manufacturing performance.

Advanced manufacturing systems operate with high precision demands for high-quality production. Usually, the precision and quality monitoring operate based on sensors with accurate precision requirements. The real-time connections between the IoT devices generate feedback between quality monitoring systems and product line systems to enhance the performance. In advanced manufacturing applications, the sensors have interconnection with separate actuators or have built in the same actuator as a device. The interconnection of these devices is based on standards protocols. These protocols are specialized and are designed for such as heterogeneous manufacturing interfaces [85], [86].

2.5.6 Logistics and Supply Chain Management

Logistics work requires the updating of reports of tracking; these reports include information about the time and location to control the tracking of shipments. Along with the IoT development, the logistic processes have massively developed to a self-ruling process via IoT applications to satisfy the logistics demands. The IoT-based logistics applications have been developed to track the shipments in real-time and update the data. Another trend of the IoT-based system in the supply chain is the tags of barcode or RFID; the tag reader sends the data to the logistics data centre. The data is transmitted through many wireless communication technologies such as Wireless Sensor Networks (WSNs), GSM networks, etc. The IoT solutions in supply chain management are very hopeful since many of the operators have implemented comprehensive supply chain protocols. These protocols have been implemented according to the corresponding application, e.g. the store or supermarket chain management was implemented based on a standard related database, which provides convenience when dealing with a huge number of items, or “things” [87], [88], [89].

2.5.7 Transportation

Transportation is considered an indicator of the urbanism of the countries, such that the governments have conducted extensive research to solve transportation problems and propose further comfortable applications for their citizens [90]. The IoT produces various useful services and applications for transportation and road

monitoring; the IoT also collects required data about the road situations and traffic to find the route and clarify its properties, before then transferring such information into the smart phone's application. These applications support the driver in terms of selection of highways, traffic jam avoidance, and find an empty parking space, in addition to safety of the road [91], [90]. These IoT applications are not only important for road monitoring but are considered economic as they reduce fuel consumption and air pollution. Many of the governments have sponsored research studies on IoT traffic systems to monitor the performance of their transportation and invent smart roads [92], [60].

2.6 IoT challenges

The information and communication technology (ICT) sector accounts for 5% of the worldwide electricity consumption [93]; this statistic is for operational consumption and does not consider the manufacturing. ICT power consumption includes the three main contributors: data centres, user equipment, and networking infrastructures. Cisco statistics reports have revealed that the global traffic will increase three times from 2015 to 2020, which requires a huge expansion of the networking infrastructure and capacity [94]; based on that, there are many challenges that can be identified in the deployment of the future internet. It is very difficult to draw the requirements and corresponding challenges for the complex system, such as innovation of the future internet, but we can categorise the future internet challenges into three areas: network, service, and contents. Most researchers discuss the economic aspect as an important challenge faced by the three mentioned areas. The innovation of the IoT is based on the simple premise of resource sharing and enhancing variant services, such that the IoT is becoming increasingly important when we are introducing or improving the paradigm of the future internet; indeed, the IoT is considered an important contributor in the design of the future internet [11], [95], [96]. Energy efficiency is considered one of the challenges in the IoT, alongside other challenges such as addressing, security, availability, reliability, mobility, performance…etc. Researchers are focusing on methodologies to maximise the utilisation of the IoT devices and minimise power consumption; in such as cases, the sensors will keep continuing operating for longer periods. This section highlights several challenges and the methodologies that have been proposed

for the IoT. These methodologies are considered as tools that assist IoT operators in enhancing IoT performance and Quality of Services (QoS) and reduce energy consumption [8], [39], [84], [97], [98].

2.6.1 Energy Efficiency Challenge and Solution for IoT networks

The IoT has various emerging applications to enhance services and applications, but it is very important to take into consideration that the IoT refers to a network which includes intelligent devices, and which operates independently with its sensing, actuation, computation and communication abilities [99], [100]; therefore, the energy consumption is considered a crucial factor of the IoT network. The energy is important to maintain the lifetime of the network, while effective communication between the IoT devices is considered an essential factor for IoT networks. The energy efficiency of IoT devices and the network is directly related to operating lifetime. The majority of the IoT devices are operated on battery or low power sources. When the battery of the IoT device is drained of its energy, it is costly and difficult to replace the battery due to short intervals and the very large number (billions) of IoT devices. To solve this problem, the IoT devices should operate at the longest intervals with their battery lifetime [29], [93], [101]. Energy efficiency is also considered an important factor when we are talking about electricity production; most of the recent energy production is not produced by renewable resources and is considered the main source of pollution and global warming due to the emission of carbon dioxide materials [102]. There are many research studies on network architecture to enhance energy efficiency; most of this work has introduced relay nodes in the IoT network, improved routing, and enhanced network topologies, thus improving the lifetime and connectivity of the IoT network [103].

2.6.1.1 Energy-efficient node selection

The authors in [104] proposed a paradigm that places the relay node according to the traffic of the network, e.g. in a high traffic area, one relay node is assigned for one sensor, while in a low data traffic area one relay node is assigned for three sensors or more. In other words, the relay node is responsible for network routing; it optimises path selection from source to destination. The relay node finds the path by considering the outstanding energy of the node; this is because the IoT network is

composed of low power devices, so that if the neighbour node is inefficient regarding energy level, then the path will be changed, and hence the remaining energy is considered a critical factor to estimate an energy-efficient path. The combination of hierarchical placement of nodes and energy balanced routing increases the network lifetime. Another concept, introduced in [36], conducts service discovery (SD) protocol for the IoT network; this protocol is capable of being energy efficient and having a high hit rate. This protocol divides the IoT network into sub-areas, each of which has one node which acts as Cluster Head (CH). The protocol achieves a higher hit data rate by multi-hopping through neighbours' CHs, or with area nodes routers that are adjusted according to the density of the smartphone distribution within the area. The large IoT network was divided into small areas, each consisting of some sensors and their elected CH. The highest level of the hierarchy consists of area routers that are responsible for multiple areas. To achieve the best hit rate, multi-hopping can be enabled either through neighbouring CHs or through area routers. Moreover, the duty cycles of the network nodes can be adjusted according to the smartphones' density in each area. Frequent tests were conducted to evaluate the efficiency of this proposed protocol and it has achieved its objectives.

2.6.1.2 Energy-efficient Cloud-based IoT Platforms

The cloud operates as a platform for IoT applications, which is the main reason why the cloud reduces power consumption by applying efficient system components. Al-Azez, in [105], introduced a Mixed Integer Linear Programming (MILP) model that proposed an energy-efficient cloud computing platform for the IoT. Said model states that the IoT network consists of four layers: Application, Service, Network, Physical layer. The physical layer consists of IoT devices, e.g. sensors, while the networking elements are located within the upper three layers. The routing of IoT traffic is controlled by Virtual Machines (VMs); these VMs are hosted by distributed mini clouds and said clouds are located within the IoT networking nodes. The model of Al-Azez et al. has been developed to optimise the number and location of the mini clouds, with constraints of reducing the total power consumption of traffic aggregation and processing to the minimum value. The model results conduct the optimal allocation of mini clouds in the IoT network, with

energy-saving of up to 36% compared to single mini clouds located at the gateway layer.

2.6.1.3 Energy-efficient sleeping time of the IoT

The energy efficiency of the IoT network is proportional to the lifetime to keep operations of the network as stated in [37]; an energy-efficient proposal for the IoT is composed of three layers: sensing and control, information processing, and presentation. This proposal is based on a methodology that predicts the sleep interval of sensors; the prediction is made according to the remaining battery level, previous usage history, and quality of information demands. The predicted value can be used to increase the utilisation of cloud resources by providing required resources when the corresponding sensors' nodes are in sleep mode. This methodology enhances the energy efficiency of all the IoT networks. In other words, this proposal swaps the sleeping IoT device with cloud resources, by predicting the maximum amount of data that can be required during the next process, and hence resources can be provided accordingly.

Another concept related to the sleeping time of radio interfaces is using the methodology of discontinuous reception/transmission (DRX/DTX), in [7]; this methodology allows the IoT devices to turn off their radio interfaces and switch into sleep mode. With such as a method, the researcher draws attention to maintaining the performance during DRX/DTX intervals; in other words, how to operate DRX/DTX to optimise energy efficiency while keeping the same level of performance. The proposal of [38] has illustrated the ability to maximise the DRX/DTX sleep periods, with guaranteed QoS. The key efficiency idea of [38] is the optimisation concerning QoS parameters and DRX/DTX configurations. The proposal result conducts the energy saving with QoS parameters of traffic bitrate, packet delay, and packet loss rate. With the further conception of sleeping time, an advanced methodology has been proposed by [39]; this proposal puts forth a modified DRX mechanism including the Quick Sleeping Indication (QSI) – a simple and energy-efficient solution for low complexity and low mobility machine type communications.

2.6.1.4 Reduced Hardware Architecture for Energy-efficient IoT

One of the methodologies used to minimise power consumption is reducing the hardware architecture of system-on-chip (SOC), in order to obtain a digital block design proposal for higher energy efficiency. The proposal of [40] has been demonstrated by synthesising into FPGA. The proposal results show that the proposal has achieved energy saving in the form of 24-15% of dynamic power reduction over the reference design. Lim et al., in [40], put forth a reduced digital hardware architecture SOC proposal in healthcare application, the main aim of which is to reduce the power consumption of the IoT sensor. The proposal results have achieved energy efficiency in comparison to the reference design, because of the removing of unnecessary peripherals from the module. The results have been estimated for the healthcare IoT node. The methodology of [40] demonstrates that unused components draw excessive power consumption.

2.6.1.5 Energy-efficient IoT based on sensing data compression

IoT has been considered as a strategic aspect that profits society in many applications, such as environment monitoring, smart traffic control, smart metering, etc. The IoT has led to a new trend, known as Smart Objects (SOs), which are defined as fundamental units of the IoT architecture; these SOs consist of sensors, which are interconnected to form wireless sensor networks (WSNs).

The authors in [106] reviewed and presented an optimization model that enhances traffic reliability and minimise the traffic power consumption of cloud-based IoT networks. The proposed model used a standby route selection scheme (SBRS) to reliability by select alternative nodes in failure cases with minimum traffic power consumption. Second, we used a desired reliability level scheme (DRLS) that considering the desired reliability level and minimizing traffic power consumption. The authors also propose a reliability-based sub-channel scheme (RBS) to avoid mitigating interference in busy reliable routes. The authors also proposed a reliability-based data compression scheme (RBDS) to conquer the limits of the capacity of the links. The results display that the average power saving of 57% in SBRS and 60% in RBDS compared to DRLS.

2.6.1.6 Energy-efficient IoT based on MAC Protocol

The proposal of [42] works on the MAC protocol and has the ability to reduce power consumption in the IoT network. This proposal focuses on the power-controlled sensor MAC protocol (PC-MAC), which is familiar with wireless sensor networks (WSN) requirements. PC-MAC has an ability to enhance the energy efficiency, due to decreasing the power of transmission and avoiding the collision. The mechanism of PC-MAC, as in [42], is to send frames with low transmission power levels instead of high levels. In addition to this, PC-MAC has the property of collision avoidance, which also means that it saves power and time by avoiding resending data. The proposal has been simulated in [42], and the results of said simulation indicate that PC-MAC protocol achieves a power saving of up to 50%-96% compared to S-MAC.

2.6.1.7 Energy-efficient IoT Data Transmission Scheme

There are many approaches, e.g. that proposed by [43], to dealing with traffic parameters, such as fault detection and error correction schemes to obtain an energy-efficient IoT network. The proposal is based on an efficient cooperative spectrum sensing (CSS) scheme, which solves the spectrum scarcity and reduces the energy consumption of the IoT network. An important factor of CSS is decision transmission between the user and the fusion centre. The energy-efficient reliable decision transmission (ERDT) methodology is proposed to reduce both the packet error and packet loss of CSS; the ERDT model increases the probability of correct decision transmission by depending on logical AND/OR rules with variant cases such as bit error rate and/or packet loss. The model results of EDRT indicate an energy saving of approximately 50% when applying ERDT in CSS.

2.6.2 Traffic delay Minimisation in IoT networks

When a source node needs to send packets to the destination or sink node, all the nodes in its transmission range between the source and destination will forward the transmitting packets. In asynchronous sleep-wake cycling networks, the nodes are randomly asleep or awake due to the energy-saving and increasing lifetime concerns; the intermediate nodes can relay the packets only when they are in the active state. We draw all those optimisation results to the time domain since low-

latency is the major objective of QoS for the recent routing algorithms, thus minimising end-to-end data delivery time; these routing algorithms have been based on the centralised and distributed routing algorithms:

- The centralised routing algorithms: in this algorithm, each node has the global network information and forwards the packets accordingly, which assists relay node selection.
- The distributed routing algorithms: in this algorithm, each node has local network information and selects the relay nodes accordingly.

There exist a plethora of research studies that have examined how to optimise or reduce the traffic delay in the IoT networks; these studies have aimed to reduce the number of hop count, node's delay, traffic media…etc. In the network design, the network is usually abstracted as a graph G = (V, E), where V stands for the vertex set or node, and E stands for edge or the communication link set of nodes. The optimisation framework then uses the objective function to obtain the target value.

The authors in [107] presented an energy-centred and QoS-aware services selection algorithm (EQSA) for IoT services composition. They proposed a model that selects the services by using a lexicographic optimisation strategy and a QoS constraints relaxation technique. The authors in [108] surveyed the recent development of SOA models for IoT and reviewed their fundamental technologies. The authors in [109] proposed a reference architecture for the smart city based on SOA concepts by integrating IoT, Cloud and Edge technologies into existing city infrastructure.

The authors in [110] surveyed the recent development of energy-efficient solutions for wireless sensors networks and reviewed some existing topologies that allow trade-offs between multiple requirements to be achieved for efficient and sustainable sensor networks. The authors in [111] presented a QoS message scheduling algorithm in IoT network-based SOA, which is more targeted towards service provisioning with the idea of service differentiation by classifying into high priority and best effort messages.

The authors in [112] surveyed the state of QoS methodologies in wireless terrestrial sensor networks to attain delay and reliability constraints in critical

applications. The authors emphasised the main challenges of implementing QoS protocols in WSN applications.

The authors in [113] introduced the paradigm of the Fog-Radio Access Network (F-RAN); the F-RAN brings the potential to meet the requirements of ultra-low-latency applications by distributing computing-intensive tasks to multiple F-RAN in the IoT nodes. The proposed paradigm migrates the efficient computing capability of the cloud to the edge of the network. The authors discussed the complex trade-off among performance, computing cost, and communication cost, with the results showing that ultra-low-latency services can be achieved by the F-RAN through a proper migration paradigm.

The authors in [114] presented a paradigm that uses Software Defined Networking (SDN) technology to manage the end-to-end IoT nodes traffic. The authors implemented their paradigm by using SDN controller to identify the traffic latency by using a probe packet over each path in the network and measuring the delivery time of each path accordingly. The proposed paradigm has the ability to discover the changes in the network topology and path delay and reroute the traffic with the objective of delaying minimisation. The paradigm results showed a reduction in traffic latency by 63% compared with the traditional shortest path routing technique.

2.6.3 Resilience Evaluation in IoT Networks

The network performance may be affected by malicious activities, disruptive cyber-attacks, Denial of Service (DoS), or any other faults which can interrupt traffic communication links even with advanced networking solutions [115]. These bring significant risks for the normal operation and services of large-scale IoT networks.

The IoT has reached a massive number of heterogeneous devices and thus is used in the deployment of a wide range of applications based on distributed open architecture [31], [56]. The researchers have proposed IoT architectures to solve the interoperability of heterogeneous systems; the proposed IoT architectures should be adapted for resilience to physical network disruption; it is also important to anticipate that many of the IoT device and links nodes will be prone to failure in the

probabilistic scheme at any time [116]. The resilient IoT architecture needs to support the semantic search, failure probability, data recovery and maintain the network dynamically and autonomously [117].

The rapid growth of the IoT applications in scale and scope produces several challenges regarding the IoT performance and dependence, one of which is emergent, where these devices are produced by different vendors to serve specific purposes, which produces a new challenge of heterogeneity [118]. Another challenge is the rapid growth of application demands; these applications involve a specific high level of security and resilience [119]. Another challenge is the growth in technology, which is illustrated by the vulnerabilities of the interconnectivity and interdependency of the devices. These challenges may lead to unexpected system failures caused by connecting IoT devices to the internet.

The resilience has an extensive consideration in various engineering, scientific, and social scopes, as it also has a great magnitude in the ultra-large-scale systems [120]. Theoretically, there are many definitions of resilience; it can be defined as the capability of a system to accomplish its operation appropriately notwithstanding disruptions and to regain its performance after a temporary system failure. In the communication systems, the adverse disruption is prospective consideration, and it is expected that the communication systems will operate even under adverse disruption and rapidly recover to their full functional services [115], [118], [119], [121], [122].

Generally, there are numerous definitions of resilience, but the most common defines resilience as the ability to operate and maintain the process with an acceptable level of service when facing various faults [123], while the network researchers in [124] defined the network resilience as the possibility of at least having a backup path within the minimum time interval, in case at least one node on the primary path has failed. In point of fact, the resilience concern aims to structure a system with fault-tolerance capabilities and implies an ability to restore from failure but does not mean that the system is very difficult to degrade [117]. Precisely, network resilience has no metric value but can be estimated using the required time that the network takes to resume its normal operation after being subjected to disruption [125]. Consequently, it is complicated to estimate network resilience in terms of the quantitative value of network resilience [115]. Another key

aspect is the number of failed nodes that the network can endure while maintaining its performance levels [125]. This leads to a probabilistic approach that assumes the network can tolerate at most n failed nodes where $1 < n < k$, which is called a $k-$ connected network [115], [126]. The idiom of k-connected network denotes that the network is preserving its nodes' connectivity after removing no more than k-1 nodes [127].

As mentioned, the IoT networks can be considered as graph G = (V, E), where V stands for the vertex set, and E stands for edge set of nodes. The vertex presents the IoT nodes with specific sensing and processing capabilities, and the edge presents the links characterised by non-negative values called the weight, such as the power consumption, distance, delay, etc.

The cardinality of the nodes and links in the network is indicated by |N| and |L|. The communication range of each node is indicated by RC(c), where RC(c) presents the maximum link's distance that a node's radio can reach. Practically, for any two distinct nodes N(c), N(d) $\in$ P, where P is set of nodes, the node N(c) can communicate with node N(d) if and only if the straight-line distance between these nodes is $|N(c) - N(d)| \leq RC(c)$. Consequently, the neighbours sub-set of node N(c) is indicated by PN(c), where PN(c) can be defined as any nodes within the communication range of node N(c), and given by:

$$PN(c) = N(d): |N(c) - N(d)| \leq RC(c), N(d) \in P \quad (2.1)$$

The number of neighbours of node N(c) is called the degree of N(c) and is denoted as D(c). The network degree is indicated by $\delta = \min(D(c))$ for all $N(c) \in P$. Mathematically, Menger's theorem describes the relationship between nodes and links of the connected network by [123]:

$$\kappa \leq \delta \leq = \frac{2 \cdot |E|}{|V|} \quad (2.2)$$

The value of κ is considered an indicator of the resilience of the network; the high value of κ points to the high level of network resilience. There are many mechanisms which are used to provide resilience in routing protocols of IoT networks, such as the Routing Protocol for Low-Power and Lossy Networks (RPL), which has recently been standardised as a routing protocol for the IoT [128], [129],

[130]; the resilience is influenced by the reliability, availability, and dependability of the system. In the traffic routing protocols, a popular technique for link failure recovery is multipath routing, where a set of multiple paths between the source and destination are selected to ensure traffic delivery. These routing protocols have advantages of high resilience and bandwidth aggregation but at the expense of higher energy consumption and traffic generation [123].

The resilience in the routing protocols can be classified into three types based on the pathfinding methodology; the first method is called proactive routing, where all paths are selected initially in the routing table, while the second method is called reactive routing, where all paths are selected on demand and updated in the routing table, and the last method is considered to be hybrid routing, which depends on both of the previous methods [124].

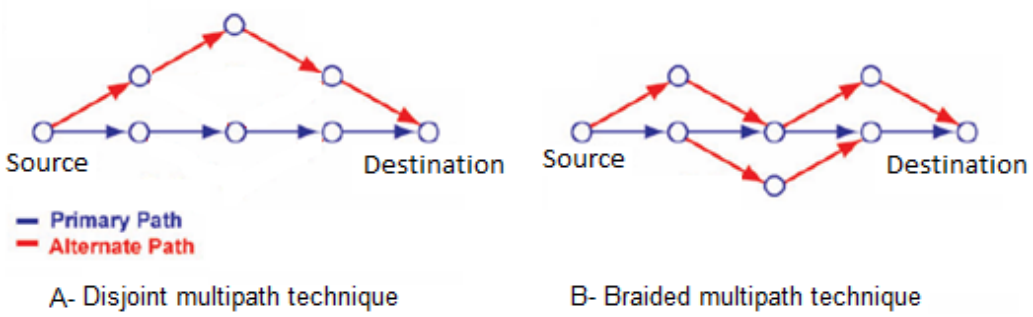


Figure 2-3: Multipath techniques.

The multipath methods have mainly two techniques to create their multipath network, as shown in Figure 2-3.A; the first is Disjoint multipath, and in this technique, multiple paths are created as an alternative to the primary path with independent nodes/links with the primary path and with other alternate paths. Consequently, a failure in any or all nodes/links on the primary path does not affect any alternative paths.

The second technique is braided multipath; in this technique, the alternative partially overlays with the primary path, as shown in Figure 2-3.B , and as a result, failure in any nodes on the primary path means that a new path discovery is required, which introduces an additional overhead [131], [132].

2.6.3.1 Retransmission-based schemes

This technique is based on setting up a multipath between the source and destination nodes; one of these paths is considered as the main or primary path to route the traffic between nodes, while the other paths are considered as alternative or backup paths; these paths are used to recover from the traffic failure of the primary path, and they are sustained by sending a "Keep-alive" signal continuously over them. When a primary path experiences failure, the intermediate node will send back the data packet to the source node and a failure report to the destination node. As a result, the source and destination nodes will remove the failed path information from the routing table and switch the traffic to the alternative path. Once the primary path is set up again, the routing table will add this path and the traffic will switch back on it. This technique has other effects on the network performance in terms of data delivery, recovery time and message overhead, memory constraint due to data caching at the source and intermediate nodes [121], [133].

2.6.3.2 Replication-based schemes

This technique fulfils the resilience requirements of traffic by sending multiple copies of the same traffic data over selected multiple paths from the source node to the destination node; thus, traffic replications are not happening only at the source node but at every intermediate node in the network. This technique has the advantage of a high delivery ratio without packet caching in the memory, and there is no need for signalling of state maintenance between the source node and destination node, because even in the case of partial data packet loss, the destination node can recover the packet from the other copies of the packets. If the destination node receives two copies of the same packet, in this case, the elimination function will ignore the extra copy and keep only one copy to upper layers. In this way, said technique achieves high resilience in terms of data delivery time but at the expense of the high energy consumption that arises due to the traffic overhead at each node along with the network [134]. The replication technique has improved the packet delivery ratio. There are various approaches that have been implemented to improve the delivery ratio in the spatial domain by routing over multipath or using parallel transmissions i.e. Packet Replication Techniques (PRT) [135].

2.7 Summary

This chapter outlined the main concepts related to IoT. It provided a general overview of the evolution of IoT and presented a review of the recent IoT architecture classifications and the layered structure of each of these architecture classifications, along with the advantages of each layer and architecture.

Furthermore, the main elements of IoT were reviewed, and a brief explanation was provided for each of these elements and its impact on an IoT system. There are a huge number of applications related to IoT, and it is difficult to summarise all of them; consequently, in this chapter, the recent IoT applications were summarised according to their domain, and the impact of IoT on these domains was highlighted.

IoT is restricted by several challenges such as those related to addressing, security, energy efficiency, availability, reliability, mobility, and performance.

Because IoT has limited power and processing resources, energy efficiency is considered one of its main challenges and thus, was addressed in this chapter. Although several solutions have been proposed for problems related to IoT, it is very difficult to draw the requirements and the corresponding challenges for a complex system such as IoT. This chapter presented a discussion of the important challenges of energy efficiency, traffic latency, and resilience faced by IoT networks.

Chapter 3

Service Virtualisation in IoT networks

3.1 Introduction

Information technologists have defined ‘virtualisation’ as a framework or methodology of dividing the resources of an infrastructure into multiple execution environments to allow multiple service providers to use these resources seamlessly and efficiently, by using hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, or other concepts or technologies [1], [2]. Virtualisation enables multiple independent users to use the physical computing resources efficiently, by abstracting them into logical units running multiple simultaneous applications [136]. It is a widespread concept in different fields. It combines various technologies for ‘virtualising everything’. To prevail in the invention race of virtualising everything, many of the leading IT companies are integrating the environment with software-enabled and internet-based businesses by relying on the competitive and economic impact of service virtualisation. Despite the significant prospects of service virtualisation, important challenges remain to be addressed to enable virtualisation deployment, e.g. technical problems such as interoperability in addition to network management and security, as well as non-technical issues such as protocol standardisation and governance regulations [137].

Virtualisation is a promising technique of IoT deployments, as multiple applications will be implemented on the same virtualised IoT network [137], [105].

In this chapter, we present a review of the existing works on service virtualisation in IoT networks and discuss several important aspects of service virtualisation: overview, architectures, framework, motivations, performance metrics, and enabling technologies. Then, we explore some broader perspectives and challenges in realising service virtualisation. Finally, we review the existing leading projects and testbeds for virtualisation in IoT.

3.2 Definition of service virtualisation

The word ‘virtual’ is expressed as an opposite of ‘real’ or can be defined in the dictionary form as an effect without a real-life appearance; accordingly, virtualisation has been applied in various domains and has been exploited in different trends [138]. Virtualisation uses the ubiquity and heterogeneity of IoT nodes to support a plurality of application domains. However, virtualisation in IoT networks has attracted significant and pertinent companies to research how numerous applications can use the same IoT infrastructure, which is known as service virtualisation [98], [139].

Information technologists have defined ‘service virtualisation’ as an operation of simulating and capturing the behaviour, data, and performance characteristics of a dependent system and deploying a virtual service that represents the dependent system without any constraints, thus allowing the applications to be delivered with fast performance, low cost, and high reliability. Service virtualisation involves combining new technologies and methodologies toward the trend of ‘virtualising everything’. Recently, software developers have bound the virtualisation technique with a developing software-enabled or internet-based product, and thus, it is applied across many industries. IoT virtualisation can be defined as a concept of building customised high-level IoT services dynamically by using real-time collected information from low-level IoT devices and creating a customised virtual service for the clients seamlessly and efficiently. Virtualised IoT services provide flexibility and scalability to build various scalable systems [140].

Concerning IoT, virtualisation can be defined more precisely as an evolving approach that enhances the aggregation of versatile heterogeneous devices, networks, and software platforms, thereby improving application development. Virtualisation in IoT finds an efficient embedded environment in sensor networks because of the sharing of resources, services, and the network. The goal of virtualisation in IoT is to implement a user’s applications to access the resources, i.e. sensor data, seamlessly and efficiently [47].

Technically, virtualisation involves integrating hardware and software on one platform, enabling the administration and sharing of resources. IoT virtualisation is

achieved by realising virtual objects into a real physical object in a framework enriched with the context by the related information. From the structural viewpoint, the applications and the end-users exploit the framework to provide high-level services [141]. The main advantages of virtualisation are significantly reducing the overall cost of equipment, decoupling functionalities from the infrastructure, facilitating the expansion of newer services and products, and enabling flexible management. IoT virtualisation motivates many applications such as smart city, healthcare, agriculture, industrial, and traffic monitoring [141], [142], [143].

3.3 Architecture of service virtualisation in IoT

Most of the existing studies on virtualisation have focused on two approaches: The first is the vertical approach of the virtualisation of the IoT architecture. This orientation has been considered because of the decoupling of the service provider from the infrastructure provider. According to the virtualisation concept, the service is not concerned with the infrastructure and the infrastructure is not assigned to a specific service, the researchers have proposed several architectures that solve the virtualisation issues in IoT.

The second approach is related to a horizontal architecture. This approach focuses on the concept of virtualisation types. Thus far, researchers have proposed node virtualisation and network virtualisation; these are also known as node-level and network-level virtualisation.

Sensor-level or node-level virtualisation is the ability of multiple applications to use a single sensor node or a set for sensors nodes to execute their tasks [139], [143], while network-level virtualisation is the ability to allow multiple service providers to apply their multiple virtual networks to a networking environment in that, isolated from the others, each virtual network shares and utilises the resources of the infrastructure’s network managed by the infrastructure providers [1], [47].

Practically, there is no typical standard paradigm for service virtualisation in IoT because of the specific problems and issues related to the use of the IoT technology. Most of the virtualisation architectures are considered for specific applications. The common key of the characteristics approach is as follows: A virtualisation layer

between the service and the infrastructure layers allows multiple applications to use the sensor resources through policies specified by the infrastructure provider.

3.3.1 Software-defined networking virtualisation in IoT

Software-defined networking (SDN) is an emerging important technology in network design [144], [145], [146]. SDN is based on the concept of isolating the control plane from the data plane of the network platform by importing the SDN controller. This controller interacts with the IoT applications through an application programmable interface (API) [147], [148], [149], translating the application's demands and making appropriate decisions in the network, as shown in Figure 3-1. On the other side, the SDN controller communicates with a lower layer presented by the network elements, i.e. switches and gateways; these switches and gateways forward data packets based on the SDN controller's orders [24]. The SDN technology enhances the flexibility of traffic routing, load balancing, and capacity utilisation, consequently mitigating the encumbrance on the network's elements and enhancing IoT networks [150]. Furthermore, the SDN controller applies service requirements to the network infrastructure, e.g. an acceptable delay, data rate, or packet loss, such that the SDN is considered an emerging virtualised IoT paradigm, orchestrating the configuration, management, provisioning, and control of traffic in the IoT networks [24], [144], [148], [151], [152], [153], [154], [155], [156].

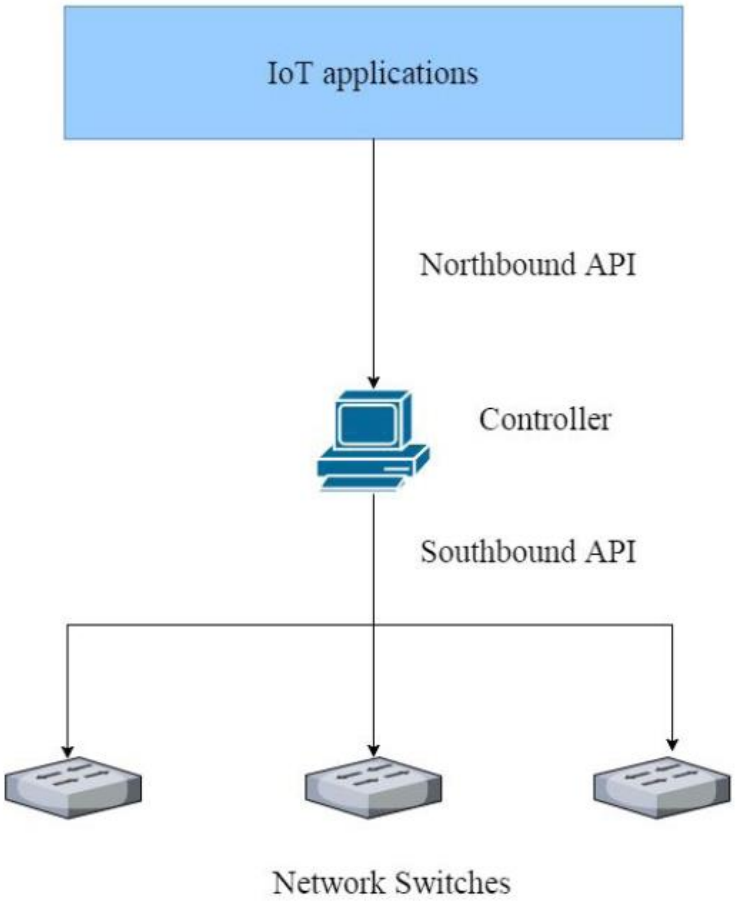


Figure 3-1: SDN-layered IoT architecture[148]

SDN has considerable advantages for the IoT network, such as centralised network management and data collection, and thus, can be applied as service virtualization technique to optimise the performance and security of IoT networks, another impact of real-time decisions based on the state and type of the received traffic [148], [149], [150], [154], [157].

3.3.2 Cognitive management virtualisation in IoT

Several presented works on service virtualisation in IoT have proposed a cognitive management framework in their design. The cognitive management mechanism is based on virtualising a real-world object (RWO) to create a virtual object (VO) that provides a service to higher-level applications. The early stage of VO realisation was based on RFID in the last decade; recently, VOs have been combined to form a composite virtual object (CVO). The CVO is considered a cognitive mash-up of semantic interoperated VOs, thus providing their services to

the applications. The main aim of the cognitive management mechanism is self-management, such as configuration, protection, learning, and optimisation, to adapt to the behaviour of the upper layer's applications [141]. The cognitive management framework for IoT enables the RWO to dynamically represent the VOs, where a VO is considered a virtual representation of RWO that dynamically changes to meet the application requirements [158].

The concept of 'cognition' refers to the decision making on the framework's stages regarding which entity should be (re)used to satisfy the application's requirements. The common architecture for a cognitive management framework is composed mainly of three layers [30], [141], [158]:

1- Service layer: This layer is considered the user interface; it is responsible for user authentication. This layer has another function, that of the translation of a user's application into the corresponding requirements and policies. The translation mechanism is achieved by mapping the application requirements to request parameters such as specific functions and policies. Another function in this layer is situation acquisition, which is responsible for the situation parameters for the services; the situation parameters are combined with the service request to specify a service property such as reliability.
2- CVO layer: Once the service layer generates a request and the situation parameters, the CVO layer searches for the available corresponding CVO that can be (re)used for the requested service. The CVO searching mechanism is achieved by request situation matching that compares the requirements with the corresponding parameters in the CVO registry and finds the appropriate CVO that meets the approximated satisfactory ratio. In case there is no matching CVO for the requested service, a CVO creation request is generated by the request situation matching to the decision-maker, and the decision-maker is responsible for composing the CVO from the VO registry.
3- VO layer: This layer is responsible of the virtual representation of the RWO; the VO may be represented by one or more than one RWO. The VO is implemented by using web services and is responsible for the contextual information because of the communication with the RWO. Each VO can communicate with the corresponding RWO directly by using web services or

indirectly through interfaces or gateways by using various communication technology standards such as machine-to-machine (M2M) and universal plug and play (UPnP). The VO layer also has a VO registry that stores the semantic information about the VOs. The VO's information, including the corresponding RWO, location, and function, in addition to the access right of the VO, has to be ready for the upper layer and the service provider at any time [62], [141], [158], [159] [160], [161], [162], [163].

3.3.3 Container-based virtualisation

Container-based virtualisation is considered a revolutionary technique in the field of IoT; this technique brings the advantages of light-weight virtualisation and management to smart objects. The container-based technique creates multiple user instances inside a specific partition of the physical objects [32], [142]. Container-based virtualisation has several advantages over other virtualisation techniques, such as fast installation, construction, and initialisation of the virtual instances, in addition to the benefit of the small memory size for the virtual image requirements, which enables the physical node to virtualise more applications and services; therefore, the container-based technique is considered well matched for the IoT scenarios [164], [165]. This technique enables the operators and developers to efficiently run their applications in low-processing-capability devices, such as the IoT and single-board computers. Technically, the container-based technique has a distinct approach for virtualisation and isolation; this approach avoids generating a high overhead due to isolating the application's process at the operating system level of the host [142].

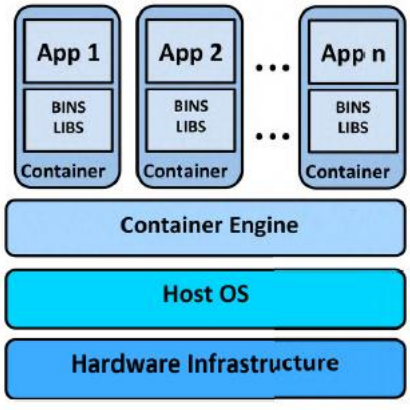


Figure 3-2: Container-based virtualisation architecture

As shown in Figure 3-2, each container instance includes the application and the corresponding demands; it operates on its own user space on the host's operating system, while the operating system is shared for all of the containers [32]. The container engine is responsible for the application container processing, such as construction, management, and removal. It can process multiple applications simultaneously and in an isolated manner. In this section, we review the recent works that have applied container-based virtualisation techniques in IoT networks [32], [142], [164], [165], [166], [167], [168].

3.3.4 Business model virtualisation

The concept of virtualisation aims to separate service from infrastructure; the business model realises this and allows the use of the resources of the physical nodes by the user's application through multiple service providers. This business model is considered a cost-efficient approach in wireless network virtualisation [47]. The present distinct architecture of business model virtualisation is layered by an application-level user (ALU), service providers (SPs), and an infrastructure provider (InP). All these layer entities are coordinated by business rules. The physical sensor's nodes are demonstrated by the InPs; InPs are aggregated on the platform of the SPs, and the main function of the SPs is to abstract services with the corresponding InPs to respond to the ALU's queries. The ALU enables the users to use the services provided by the collaborating SPs and InPs. The main advantage of business model virtualisation is the ability to effectively serve various applications enabled by the combination of multiple-service SPs and InPs. In this section, we review the recent works that have presented business model virtualisation architectures in IoT networks. Recently, a smart house application was presented that allows virtualisation in a sensor network to enhance flexibility and security; the proposed business model was evaluated by the simulation of two smart house application scenarios, and a comparison of the cost between the virtualisation and the traditional approaches [169], [170]. The proposed virtualisation architecture reduces the total cost compared with the traditional approach, which is composed of the individual nodes of both the smart house applications.

3.3.5 Service-oriented architecture-based virtualisation

IoT provides several small services, thus offered by the smart nodes. To integrate these services and generate higher-level services, there is a need for an emergent architecture consisting of an elementary service. One of the well-defined architectures is the service-oriented architecture (SOA) [153]. It has the ability to integrate heterogeneous systems efficiently and support both node-level and network-level virtualisation in IoT [100]. The SOA supports a wide range of cloud computing paradigms such as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). In addition, service-oriented network virtualisation enables a network-as-a-service (NaaS) paradigm [153]. As an architecture, the SOA is based on the concept of defining each function as an independent service with an appropriate interface that can be orchestrated in the form of a business process [48], [153]. Therefore, the SOA enables developers and organisations to construct, deploy, and integrate applications by using sets of reusable assets and services [108], [171]. The SOA paradigm is widely used for smart embedded devices. As a result of the convergence of the products and solutions, the integration with SOA will reduce the network overheads [108].

The SOA has the significant benefit of providing an interoperability paradigm with the concept of service composition as an essential constituent; service composition produces a composite service with functions that individuals cannot be provided by a single existing service [48], [108]. Additionally, the SOA supports both centralised and decentralised management approaches and allows the efficient reuse of both software and hardware [172]. Architecturally, the SOA is interposed between the application and the technical layers of the IoT in the form of middleware [47]. The middleware is generally composed of three sub-layers: the higher layer is the service composition layer that designs the workflow of the coordinated services in the form of a business process; the service management layer is responsible for the discovery, mapping, monitoring, and control of the service provided by the objects abstraction layer; and the objects abstraction layer associates objects with their corresponding functions [45], [173], [174], [175], [176], [177], [178], [179].

3.3.6 Uni-kernel virtualisation

Uni-kernels are considered a lightweight virtualisation technique because of their small footprint and flexibility. This technique has single-purpose applications in IoT and cloud applications; it is sealed against any modification after deployment; in other words, a uni-kernel is applicable to a fixed application that does not allow for incoming services to its operating system [32]. The uni-kernel paradigm provides an interface for various HW platforms that allow the developers to implement pervasive solutions in different coding languages such as MirageOS [180], HaLVM, IncludeOS [181], and ClickOS [182]. The high level of security is considered the main privilege provided by the uni-kernels; uni-kernels and security are tightly merged against any attack to their applications. Therefore, because of their direct compliance with the entire process in the application layer, there is no uniform operating layer in the uni-kernel architecture. The uni-kernel has a similar OS to that of the container-based virtualisation, but the uni-kernel has a hypervisor layer that separates each application as a stand-alone entity with its corresponding characteristics, i.e. separate file systems, independent process space, and independent virtual network interfaces [13], [32], [183].

3.3.7 Resource-oriented architecture virtualisation

The resource-oriented architecture (ROA) is considered a middleware design that deals with the device as a resource rather than the service and provides direct accessibility through a web architecture, such as URI, HTTP, or REST. The ROA enables the users to access the resources directly and efficiently [47], [112], [177]. The ROA is widely used in node-level virtualisation because of its simplicity and low overhead for the constrained devices; the ROA is preferred in representational state transfer (REST) web services, where the system entities could be created, read, deleted, and updated by hosting a web service [67]. The REST is a globally accepted technology that supports IoT applications, and the REST proposal is based on the cooperation and sharing of resources. The REST provides a set of available predefined operations, where each operation is unique and limited to all the resources, which explains the interoperation between ROA and REST [137].

Architecturally, the IoT devices, as resources, are communicable through a physical module and a standard common interface; this interface is a set of stateless

operations, such as REST in IoT web services. The system that complies with the REST standard is called the RESTful system [184]. The IoT resources are orchestrated by the provided RESTful application and accessed through uniform resource identifiers (URIs). Although the representation protocol stacks are complex to resource discovery, publication, invocation, etc., the ROA enables various representations of resources through various URIs [46], [185]. The ROA supports the trend of the Web of Things (WoT) [57]. The WoT innovates new approaches for accessing smart devices through web-based approaches and using these devices in IoT applications[46], [185], [186].

3.4 Advantages of virtualisation in IoT

Virtualisation in the IoT network creates an environment of embedded sensor networks; thus, virtualisation is considered an emerging approach that aggregates multiple heterogeneous nodes, enhances software platforms, and innovates applications.

3.4.1 New services to the cloud

Virtualisation in IoT has led to new trends toward the coupling between IoT and cloud approaches, such as sensor-as-a-service (SenaaS). This approach enhances the efficient creation, management, discovery, and delivery of a node's function [175]. Another trend envisioned by the virtualisation in IoT is sensing and actuation as a service (SAaaS) [187]. These trends are considered essential approaches towards infrastructure-as-a-service (IaaS), bringing flexibility and scalability to IoT and cloud applications [188].

3.4.2 Energy efficiency

IoT nodes are based on low-power microcontroller and low-power wireless communication concepts with limited power resources such as small batteries. The low power or energy efficient solutions are considered the significant design and implementation factors. Recent virtualisation techniques have introduced new ways to reduce energy consumption through node selection, task scheduling, routing capability, and other energy-aware designs. Many researchers have used virtualisation approaches for minimising the power consumption of IoT networks or

using the IoT for reducing the power consumption of the current projects such as smart grids [105].

3.4.3 Reducing cost and hardware overhead

Virtualisation is based on the sharing and re-use of resources so that the overall cost of hardware and management can be reduced because of the multi-use of the infrastructure with HW utilisation constraints [155]. Virtualisation has not only reduced the capital expenditure (CAPEX) of the infrastructure but also reduced the operational expenditure by reducing the traffic cost [143]. Virtualisation in IoT networks has simplified the complexity of any application, even the single-purpose ones, without overlying on an additional individual network, thus reducing the total cost and HW involvement [87], [148].

3.4.4 Flexibility

Several researchers have presented virtualisation in IoT paradigms with a high level of flexibility and adaptability. This flexibility is simply defined as the system's ability to adapt to any internal or external changes and not fail. Virtualisation in IoT provides a high level of flexibility to the service providers to deploy their applications [159]. Virtualisation in IoT enables a certain level of flexibility with traffic routing in the virtualised environment [145]. This flexibility is a significant parameter in the QoS of the network [49].

3.4.5 Availability

Availability can be defined as the probability of the system to be available and ready for use. For IoT, availability is considered a strict demand, particularly for assisted living and healthcare applications [189], [190]. The most important property of virtualisation is that it can keep a virtualised service running even though the physical resource has to be upgraded or shutdown, by migrating the virtual service from the current physical resource to another one, and migrating it back when the original resource is ready [166]. Moreover, virtualisation has a monitoring function that checks the status of the virtual sensor and the corresponding sensor in the infrastructure [191].

3.4.6 Scalability

Scalability in IoT can be defined as the ability of adding, upgrading, or removing nodes over time depending on the required system capacity. Virtualisation facilitates the addition of a new IoT node with its basic service to the network (i.e. add and register node into the cluster head) because of the ability of service discovery in the virtualisation. Therefore, the infrastructure can extend its ability to serve more applications requested by the service providers [166]. Scalability is not only limited to the infrastructure, but it has produced new types of service and service providers also due to supporting various user's applications and service provision [30]. Scalability has been studied widely because of the limitation of the IoT infrastructure and the physical system to adapt to the service requirements [32].

3.4.7 Simplified managed services architecture

The virtualisation concept is based on the decoupling of the service provider from the infrastructure provider, thus providing better opportunities for all providers to focus on their growth strategy, enhance service delivery, and satisfy customer requirements; consequently, virtualisation uses a trust-managed architecture for both the entities to serve all the applications and networks [33], [99], [100], [139].

3.4.8 Data fusion and aggregation

The main objective of the IoT is to facilitate the interaction among smart objects and between users and smart objects. The IoT objects collect information for frequent use by specific applications. From the perspective of information technology, the virtualisation layer is not responsible only for the information and service provision to the user but also for the aggregation of data from the IoT nodes [47], [192]. The fusion of the information from the embedded IoT nodes can be used by various applications through context-aware virtualisation [159], [193]. This approach enables the reuse of information rather than that of the nodes; the information can be used, managed, combined, fused, and claimed to enhance the dynamicity and scalability of the applications [194].

3.4.9 Load balancing

Load balancing can be defined as the equalising of the workload among all the nodes and/or links in the network. The workload can be expressed in terms of the number of processes and processing time and is based on other measurements such as processing resources, architecture, and capacity. The virtualisation in IoT enables dynamic load balancing in the underlying physical infrastructure of the IoT resources [195]. Such virtualisation enhances smart routing of traffic in the IoT network because of the load balancing in the links based on their capacity utilisation [196], [197]. The virtualisation in IoT allows for various types of load balancing, such as network-based, inter-cluster-based, intra-cluster-based, sensor-based, and applications-based [47], [50].

3.4.10 Decoupling and isolation

One of the essential virtualisation features is the decoupling of components, such as hardware devices, software modules, applications, communication radios, services, and networks, from one another. Virtualisation enables the logical isolation of these entities to enhance the development of each approach individually, in addition to increasing the security and adaptability levels. The isolation of the components (i.e. software from hardware) enables the software to evolve from the hardware independently. The same is true for the other components [198], [199].

3.4.11 Abstraction of heterogeneity

IoT networks can contain a vast number of diverse objects/devices with various types of communication modules that support various users/stakeholders. To ensure a high level of integrity between the heterogeneous components of the IoT to produce an application properly, virtualisation allows the abstraction of the functionalities and the underlying infrastructure, and the abstraction feature of virtualization provides a solution for the technological heterogeneity to address the integration between the hardware and the software, and resolve the interoperability issue of the different stakeholders of IoT systems [141], [153], [194], [200].

3.4.12 Reliability

System reliability is considered an important factor in system performance and can be defined as the probability of producing the output correctly over time [201], [190]. In IoT networks, data reliability has considerable importance from aspects such as energy efficiency and availability [134]. Virtualisation techniques provide two approaches for data reliability: The first approach is based on the direct data transformation to the sensor's nodes; this is considered to be a low-reliability-but-energy-efficient approach. The second approach is based on the data transfer from a sensor to the sink node or a gateway as a powerful device in the network; this approach has higher reliability but consumes more energy because of the high network overhead [47] [112].

3.4.13 Application adaptability

Adaptability can be defined as the ability of a system to efficiently and rapidly adjust to the circumstances by itself. An adaptive system is, therefore, an open system that can alter its behaviour according to changes in its environment or in parts of the system itself. The IoT provides the adaptability property to objects where smart interactions between objects that adapt to the current situation without any human involvement become the next logical step to people staying connected anytime and anywhere [18].

The IoT has a vast range of applications such as smart home, smart city, healthcare, military, and traffic control. These applications have distinct demands such as those for more reliability, stability, availability, security, and/or real-time networks. Because of the variety of demands and the resource limitations of the IoT networks, it is quite possible that the performance of the various running applications cannot be accomplished seamlessly [19]. Virtualisation provides a comprehensive environment to the developers to deploy their applications. In addition, it provides the ability to run concurrent applications on multiple networks consisting of variable resources of constrained devices and a wide range of properties. Therefore, virtualisation can manage multi threads of adaptability to fulfil the application needs [8], [47], [69].

3.5 Challenges and open issues of virtualisation in IoT networks

Virtualisation in IoT networks has many challenges and open issues still pending. The following challenges need to be studied by researchers and considered to be future work directions.

3.5.1 Security, privacy, and trust

Security is considered an essential factor of virtualisation in IoT. In the virtual plan, an acceptable level of security is determined in the communication between the non-constrained virtual nodes in a virtual network; eventually, secure communication between the resource-constrained IoT nodes and non-constrained devices in a physical network remains a significant challenge. Several researchers have presented virtualisation techniques thus require low processing resources and intend to implement secure communication between sensors and between sensors and non-constrained devices through lightweight encryption. Two other challenges are the privacy of humans and the confidentiality of information; despite many privacy approaches recently applied, the pending challenge is to produce fast encryption, less power consumption, and an efficient privacy scheme for a virtualised environment[166].

3.5.2 Managing heterogeneity

Virtualisation is responsible for the communication between the applications and the hardware with an efficient management of the services and the infrastructure. The considerable variety of service demands and infrastructure resources has resulted in the challenge of extreme heterogeneity. Despite the fact that the increase in the number of connected IoT devices has been properly studied in many research projects, the heterogeneity and its corresponding demands are still open questions, along with the introduction of a unified structure for all the platform types [148], [202], [203].

3.5.3 Orchestration and monitoring

Virtualisation is not only responsible for virtualising a physical object with the virtual demands, but also responsible for the trade-off among the various QoS dimensions of the multiple applications. Therefore, virtualisation techniques attempt to control and monitor various QoS parameters and the available resources according to the trade-off policy between them. Moreover, virtualisation must fulfil the application demands collectively, which is considered one of the main challenges of virtualisation in IoT design. An example of the pending issues is to design an energy-aware system with QoS policies in one virtualization paradigm; this design must reduce the power consumption of the multiple physical resources and fulfil the virtual network's demands. However, QoS should not be compromised. In addition to the mentioned challenges, other metrics pose additional challenges, such as the mobility of IoT nodes; this poses many challenges such as varying power consumption, QoS provision, clustering, and localisation [47].

3.5.4 Elasticity in service provisioning

Service provisioning in virtualisation needs prior operations such as service discovery and publication processes. The IoT nodes are considered resource-constrained devices; thus, the dynamic publication and discovery mechanisms are considered to be one of the virtualisation challenges in IoT networks [204].

3.5.5 Redundancy

Redundancy can be defined practically as an implication of extra elements that are not involved in the available operations but are required in the event of a failure of other elements. The redundancy in virtualisation must be eliminated for limited operations in IoT networks such as localisation and clustering techniques for a virtualisation process, in addition to the efficient algorithms for delay tolerance, load balancing, and error minimisation for the virtualised services. All these operations increase the processing overhead. The conventional redundancy approaches will not be efficient in the virtualised IoT environment [139], [169], [205].

3.5.6 Standards and regulations

A standard can be defined as a supportive design for a wide range of applications that satisfies their requirements. The recent virtualisation in IoT techniques have their own individual standards that operate with various services and resources because of multiple modules. The service provider must follow the parametric standard format of the services and resources from distinct heterogeneous networks. The standards should be designed using multiple entities in the networks, which is considered a significant improvement [206], [207]. Such standardisation presents an open challenge for virtualisation in the IoT network because of the lack of uniform and consensus platforms for the software and the hardware, and the continuous growth of the IoT service and the increase in the resource heterogeneity. The virtualisation in IoT should standardise the widespread IoT networks by merging distinct nodes, providing mechanisms for data aggregation, and dealing with the integration problem of heterogeneity [145].

3.5.7 Data storage and processing capability

The IoT nodes are considered tiny devices with small batteries, limited processing, and memory capabilities; thus, these nodes are prone to failures on their operations because of the software and/or the hardware. The open challenge for the developers is to design lightweight virtualisation with a minimal effect on network performance. Such virtualisation demands more memory for service discovery and monitoring, in addition to the decoupling between services and infrastructure at an optimum level with significant improvement for virtualisation in IoT networks [142], [207].

3.5.8 Industrial readiness and prospects

The IoT trend is considered a cornerstone of the future Internet and has had a significant impact in the industrial sector. Further, the emergent approach of virtualisation in IoT is in the face of the rapid development of communication technologies. One of the main challenges in the industrial sector is that IoT is developing faster than the standards and regulations, thus producing various approaches and paradigms, creating an intractable task exacerbated by the heterogeneity of the involved technologies. The industrial sector seeks to unify the

precision of the virtualisation techniques to deploy a wide range of IoT scenarios and to ensure the adaptation of the virtualisation of IoT with other network technologies [32].

3.5.9 Physical layer issues

Virtualisation is based on the concept of exploiting the infrastructure presented by the physical network resources, physical resources are used by multiple service providers simultaneously and seamlessly, thus demanding well-defined interfaces to enable the interoperability between the infrastructure resources. These resources pose a few physical layer challenges such as function selection, generation of the carrier frequency, modulation and demodulation, encryption and decryption, data transmission and reception, radio interferences [208]. Another challenge is resource allocation. The virtualisation in the IoT network requires a dynamic allocation and discovery of the resources presented by the IoT nodes and links [139], [149].

3.5.10 Framework management and protocol portability

Virtualisation in IoT requires the execution of multiple tasks requested by at least one service provider in the infrastructure, but because of the wide diversity and heterogeneity of IoT approaches and with the absence of widely accepted standard protocols, many vendors have been encouraged to develop proprietary protocols to run inside their sensor networks. It is unrealistic to develop and support a protocol handler for every IoT scenario and its variations. This leads us to exploit methods that use less prior knowledge and to try to extract the model of the services automatically, adapting service virtualisation for IoT [206].

In this theses, we present a framework to solve management issues, orchestrating and monitoring, and redundancy challenges with contributions of enhancing the energy efficiency, traffic latency, and resilience.

3.6 Summary

Many researchers have provided solutions for IoT. These efforts have initiated platforms for designing, deploying, setting up, and then maintaining complex tasks to be used as testbeds. This review describes some of the most currently active and

meaningful service virtualisation techniques for IoT. Here, we discussed their architecture and characteristics. We found that service virtualisation is more heterogeneous and multi-purpose and has several advantages. It offers an increasing set of services for a large range of user applications and demands. We listed the widely used testbeds and awarded projects, which serve as tools for IoT developments and experimentation.

The technological trend of service virtualisation is to integrate and modify the services in a new order; this has resulted in several challenges. These technological challenges and open issues have been summarised in this review.

Chapter 4

Service Embedding in Smart Buildings with energy efficiency and latency minimisation

4.1 Introduction

The decentralised and heterogeneous properties of IoT devices require an efficient architecture that hides such heterogeneity from higher level applications and provides interoperability for information exchange with other IoT devices [8]. Service Oriented Architecture (SOA) is considered as a viable middleware between user's applications and the IoT physical layer that support interoperability between those heterogeneous IoT devices [9]. This chapter aims to evaluate the energy efficiency and traffic latency of embedding service requests in IoT networks in building setting implemented following the SOA. Building a framework with SOA based middleware, enable both of node and network virtualization in addition to other advanced properties of SOA.

This chapter introduces a generic MILP model that has been developed to minimize the power consumption due to both processing and the traffic flow through the network. We apply this model to simulate a smart building setting. We formulate the problem of finding the optimal set of IoT nodes and links to embed BPs into the IoT layer considering three objective functions:

i) minimising network and processing power consumption only
ii) minimising mean traffic latency only
iii) minimising a weighted combination of power consumption and traffic latency.

This problem is formulated using Mixed Integer Liner Programming (MILP).

4.2 Proposed Architecture

In the smart building setting, many services employ IoT nodes such as:

- Security services employing motion detectors, RFID, display screens and alarms.
- Energy saving services employing motion detection, temperature sensors
- Fire protection services employing temperature sensors, smoke detectors, water sprinklers and alarms.
- Entertainment services employing noise detectors, and temperature sensors.
- Administration services employing motion detectors, temperature sensors, door actuators, and alarms.

These services and other services can share the same sensing and actuating facilities like sensors for motion, temperature, sound, smoke detectors in addition to the processing modules of the IoT nodes. The IoT can provide multiple services but it requires an efficient architecture that hides such heterogeneity from higher level services and provides interoperability for information exchange with other IoT devices. The SOA enables the abstraction of the IoT node functions to be translated into basic services which in turn can be composed into complex services and exploited by the upper application layer.

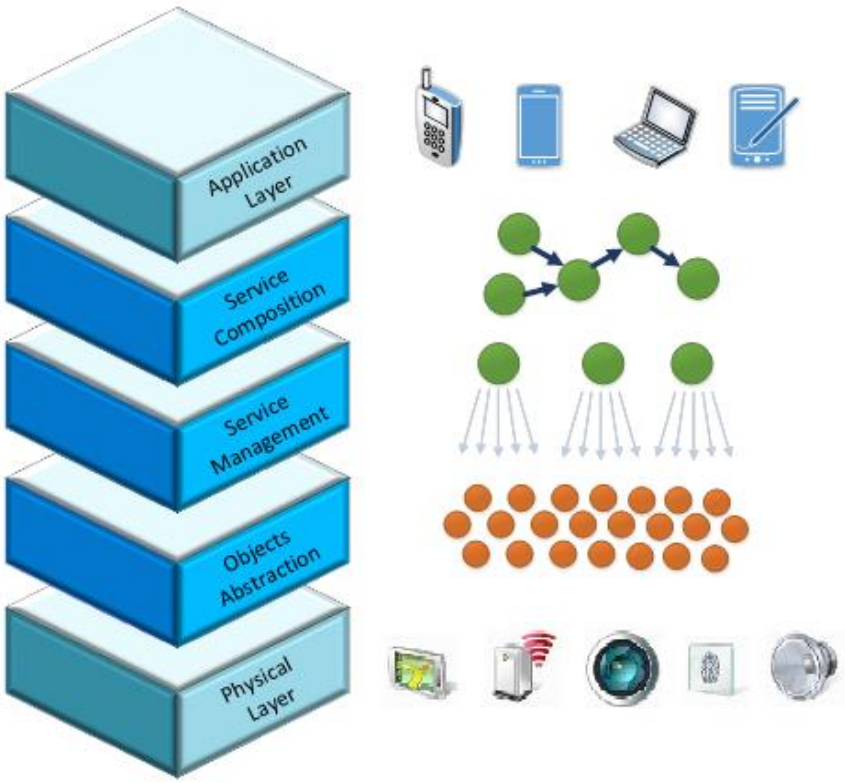


Figure 4-1: SOA-based middleware architecture for the IoT.

Figure 4-1 depicts the SOA middleware for IoT which is composed of three sub-layers [1], [2], [10]:

- Objects abstraction layer that enables IoT devices to provide their functions as basic services to the upper layers,
- Service management layer, which is responsible for dynamic object discovery, status monitoring available services of the IoT nodes.
- Service composition layer which is where complex services are requested in the form of business processes (BPs). It describes the workflow of the connected basic services.

We develop a framework to embed service requests into a substrate network of IoT nodes. These requests are implemented following the SOA in the form of a BP. A BP is a virtual topology that consists of virtual nodes and links. The virtual nodes encapsulate the requested processing demand, sensing/actuating functions. The virtual links carry traffic between virtual nodes. The embedding process maps the virtual nodes and virtual links of each BP into nodes and links of the IoT layer.

Each BP is defined as a set of virtual nodes and links. Each virtual node has a function that requires processing and memory. Virtual nodes need to be embedded in a certain geographical zone. Virtual links carry traffic demands between virtual nodes.

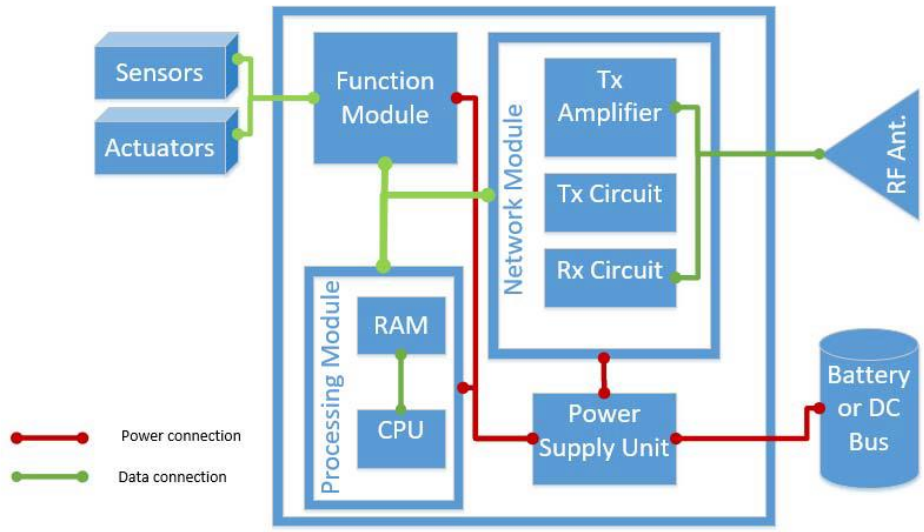


Figure 4-2: Block diagram of IoT Node.

Each IoT node is characterised by the following modules as shown in Figure 4-2:

- A processing module hosting a CPU and RAM.
- A network module hosting a wireless traffic transceiver (Tx/Rx circuit and a Tx power amplifier).
- A function module that provides interfaces to a set of supported sensors and actuators.

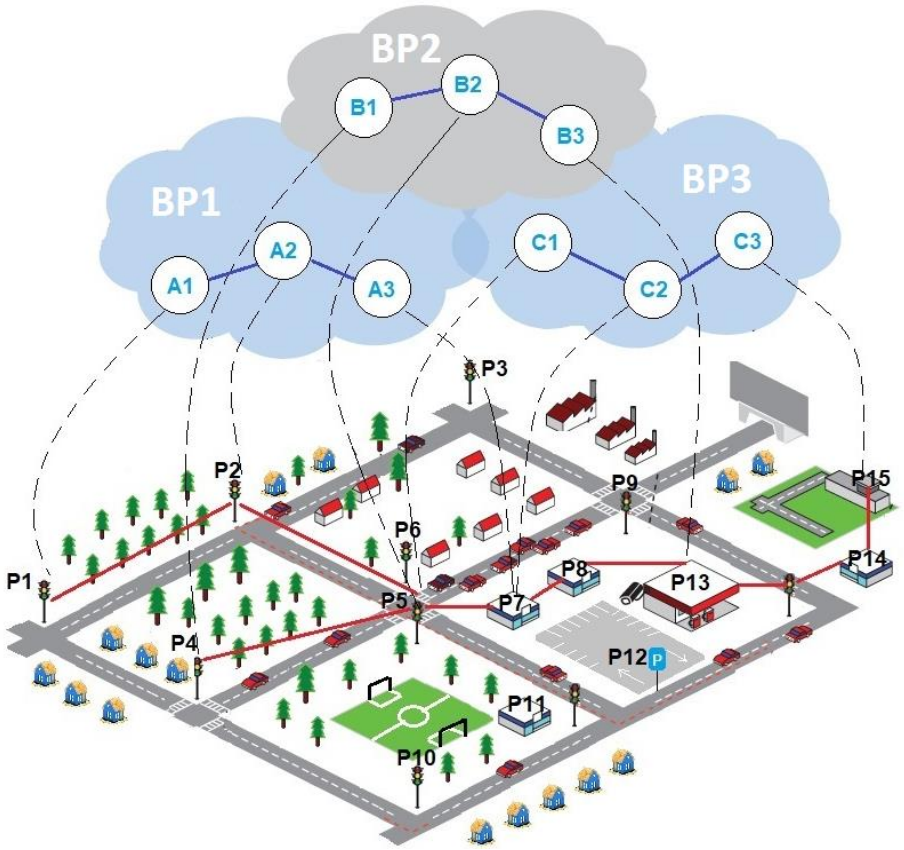


Figure 4-3: Service embedding layers in IoT networks.

Figure 4-3 gives an example of embedding two BPs. The framework embeds the virtual nodes of BP1 (A1-A2-A3) in the physical IoT nodes (P1-P2-P7), respectively; and chooses the path (P1-P2-P5-P7) to link the embedding IoT nodes. Each virtual node is embedded into an IoT node that satisfies the virtual node's requirements. An IoT node that embeds a certain virtual node of a certain BP can at the same time work as a relay node for the traffic associated with another BP. This is shown in the second embedding example where IoT node P5 which is an embedding

node for BP2 and at the same time works as a relay node for the traffic associated with BP1.

We consider a typical IoT setting where the power consumption of IoT nodes is mainly attributed to the processing and network modules while the sensing and actuating modules are externally powered i.e. the alarm and door locker is supplied by external power lines and separated from the IoT node power supply.

As the traffic between IoT nodes is routed via a multi-hop network, we consider the queuing and transmission latency which dominates over the propagation delay [1] as a network performance metric referred to it as traffic mean latency.

4.3 A framework of service embedding in IoT networks

To study the power consumption and traffic mean delay resulting from embedding BPs into the IoT network, we formulate the embedding problem as a MILP model with consideration of three different objective functions:

- Minimising the total power consumption.
- Minimising traffic mean latency.
- Minimising both total power consumption and traffic mean latency in multi-objective manner.

4.3.1 Framework Definitions

Before we give these objective functions and the constraint the embedding of BPs, we introduce the sets, parameters and variables used in the formulations:

Sets

B	Set of business processes (BPs) in the virtual layer
V	Set of virtual nodes in each BP
VN_{ia}	Set of neighbours of each virtual node in each BP ($i \in B, a \in V$)
P	Set of IoT nodes in the physical layer
PN_c	Set of neighbours of IoT nodes ($c \in P$)
F	Set of functions supported by IoT nodes

Z	Set of zones in the IoT physical layer
λ	Set of arrival rates
W_j	Set of traffic mean latency per arrival rate ($j \in \lambda$) in ms per packet

Parameters

V_{ian}^{FUNC}	$V_{ian}^{FUNC} = 1$ If virtual node a in BP i requires the function n, $V_{ian}^{FUNC} = 0$ otherwise
V_{iaz}^{ZONE}	$V_{iaz}^{ZONE} = 1$ If virtual node a in BP i requires zone z, $V_{iaz}^{ZONE} = 0$ otherwise
V_{ia}^{MCU}	Processing requirement of the virtual node a in BP i in MHz
V_{ia}^{RAM}	Memory requirement of the virtual node a in BP i in kB
V_{iab}^{TRFIC}	Traffic demand between the virtual node pair (a, b) in $BP\ i$ in kb/s
P_{cn}^{FUNC}	$P_{cn}^{FUNC} = 1$ If IoT node c can provide the function n, $P_{cn}^{FUNC} = 0$ otherwise.
P_{cz}^{ZONE}	$P_{cz}^{ZONE} = 1$ If the IoT node c is located in zone z, $P_{cz}^{ZONE} = 0$ otherwise.
P_c^{MCU}	Processing capability of the IoT node c in MHz.
P_c^{RAM}	Memory capability of the IoT node c in kB.
P_{ef}^{DIST}	Distance between the neighbouring IoT node pair (e, f) in meters.
P_c^{IDLECP}	Idle processor power in each IoT node c in mW.
P_c^{MAXCP}	Maximum processor power consumption in each IoT node c in mW.
P_c^{IDLETP}	Idle network power consumption in each IoT node c in mW.
E_{ef}^{PBT}	Energy per bit for each IoT link (e, f) in mW/kbps.
M	Large number ($= 10^8$).
P_e^{CAPT}	Link capacity for each IoT node (e) in kbps.
F_{ef}^{TR}	Transmit amplifier factor for each IoT link (e, f) in mW/kbps/m^2.

Variables

I_{iac}^{NE}	$I_{iac}^{NE} = 1$ If virtual node a in BP i has been embedded in IoT node c, $I_{iac}^{NE} = 0$ otherwise.
I_{iacn}^{F}	$I_{iacn}^{FI} = 1$ If IoT node c has the function n required by virtual node a in BP i, $I_{iacn}^{FI} = 0$ otherwise.
I_{iacz}^{Z}	$I_{iacz}^{ZI} = 1$ If IoT node c is located in zone z required by virtual node a

	in BP i,$I_{iacz}^{ZI} = 0$ otherwise.
I_{iabcd}^{LE}	$I_{iabcd}^{LE} = 1$ If the neighbouring virtual nodes (a, b) in BP i have been embedded in IoT nodes (c, d), $I_{iabcd}^{LE} = 0$ otherwise.
X_{iabcd}^{XOR}	Dummy binary variable
R_{cd}^{TRFP}	Embedded traffic demand between IoT nodes (c, d) in kbps.
R_{cdef}^{ROUTE}	Traffic between IoT nodes (c, d) traversing the neighbouring IoT nodes (e, f) in kbps.
I_{cdef}^{R}	$I_{cdef}^{R} = 1$ If the traffic demand between IoT nodes (c, d) traverses neighbouring IoT nodes(e, f), $I_{cdef}^{R} = 0$ otherwise.
R_{ef}^{TRFL}	Traffic between neighbouring IoT nodes (e, f) in kbps.
R_{f}^{TRFN}	Arrival rate of IoT nodes (f) in kbps.
$LI_{f\ j}^{\text{Lmbda}}$	Lambda indicator for each IoT node (f) ; $(j) LI_{f\ j}^{\text{Lmbda}} = 1$ if the arrival rate is (j), it is 0 otherwise.
W_{f}^{NODE}	Traffic mean latency for each node (f).
I_{c}^{PM}	$I_{c}^{PM} = 1$ If the processing module of IoT node c is powered on, $I_{c}^{PM} = 0$ otherwise.
I_{c}^{TM}	$I_{c}^{TM} = 1$ If the network module of IoT node c is powered on, $I_{c}^{TM} = 0$ otherwise.
TPP	Total processing power in the IoT network in mW.
TNP	Total network power in the IoT network in mW.
TL	Total traffic mean latency in ms.

4.3.2 Energy efficient service embedding

This embedding scenario has an objective function whose goal is to minimise the total power consumption as follows:

Objective: ***minimise*** TNP+TPP (4.1)

where TPP is total processing power and given by:

$$TPP = \sum_{c \in P} I_c^{PM} \cdot P_c^{IDLECP} + \sum_{c \in P} \sum_{i \in B} \sum_{a \in V} I_{iac}^{NE} \cdot P_c^{MAXCP} \cdot \frac{V_{ia}^{MCU}}{P_c^{MCU}} \quad (4.2)$$

where I_c^{PM} is a binary variable that indicates the activity of the processing module in IoT node *c* , P_c^{IDLECP} is the idle processing power parameter of IoT node *c* in mW, I_{iac}^{NE} is a binary variable that indicates if a virtual node a in BP *i* has been embedded in IoT node *c*, P_c^{MAXCP} is a parameter that gives the maximum CPU power consumption in each IoT node *c* in mW, V_{ia}^{MCU} is a parameter whose value gives the processing requirement of the virtual node *a* in BP *i* in MHz, and P_c^{MCU} is a parameter that specifies the processing capability of the IoT node *c* in MHz. The processing power consumption is considered to follow a liner profile versus the load with an idle power consumption. The total traffic power of the network, TNP, and given by:

$$TNP = \sum_{e \in P} I_e^{TM} \cdot P_e^{IDLETP} + 2 \cdot \sum_{e \in P} \sum_{f \in PN_e} R_{ef}^{TRFIC} \cdot E_{ef}^{PBT} + \sum_{e \in P} \sum_{f \in PN_e} R_{ef}^{TRFIC} \cdot (P_{ef}^{DIST})^2 \cdot F_{ef}^{TR} \quad (4.3)$$

where *f* is neighbour IoT node of e and is included in PN_e , PN_e is the neighbours subset of IoT node *e* , I_e^{TM} is a binary variable that indicates the activity of the network module in the IoT node, P_e^{IDLETP} is the idle network power parameter of IoT node *e* , R_{ef}^{TRFIC} is a variable that specifies the traffic between neighbouring IoT nodes *e* and f in kbps, E_{ef}^{PBT} is a parameter that gives the energy per bit for each IoT link *e, f* in mW/kbps, P_{ef}^{DIST} is a parameter that specifies the distance between the neighbouring IoT nodes pair (*e, f)* in meters, and F_{ef}^{TR} is the transmit amplifier factor [18] for each IoT link *e, f* in mW/kbps/m^2.

The network power consumption is a function of the traffic and distance between the source and destination nodes. The network power consumption of each link consists of the idle power, the power consumed per bit by the electronics in the

transmitter and the receiver, and the transmitter amplifier power consumption which is calculated based on the radio energy needed based on Friis free-space equation in our setting (note that higher propagation factors beyond Friis square law, e.g. cubic or higher, can be considered, and are a straight forward extension of our equations, but are not considered here) [64], [209].

4.3.3 Low latency service embedding

The second scenario in our framework is concerned with minimising the total traffic mean latency of the service embedding. The framework minimises the traffic mean latency in the IoT network using the following objective function:

Objective: ***minimise*** TL (4.4)

where TL P^{TL} is the total traffic mean latency in the network given by:

$$TL = \sum_{f \in P} W_f^{NODE} \quad (4.5)$$

Our network is modelled as an open Jackson network of multiple M/M/1 queues where the utilisation is less than 1 at every queue [210]. For simplicity, we consider each node as M/M/1. The M/M/1 model refers to a system with a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution as shown in Figure 4-4.

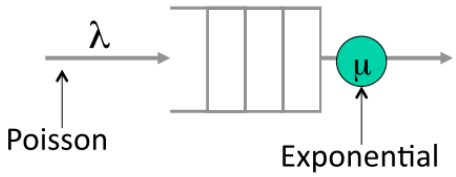


Figure 4-4: Single server queuing system.

The mean latency is the average time that the packet takes to pass through queue and server, which is given by:

$$W_f^{NODE} = \frac{1}{(\mu_f^{NODE} - \lambda_f^{NODE})} \quad (4.6)$$

The arrival rate represents the average rate of successful packets transfer to the node through physical links per time unit. Mathematically, the arrival rate is the summation of data rates delivered to the node in the network.

In our framework, we considered that the service rate μ_f^{NODE} is fixed for each IoT nodes in the network. The service rate is the transmission rate of the network module. A variable, λ_f^{NODE}, is created to calculate the summation of packet arrival at each IoT device.

Since we are using linear programming, equation (4. 6) must be converted to a linear format. To facilitate this, we use a lookup table indexed variable to calculate the traffic mean latency. The lookup table indexed-variables method depends on generating lambda indicator as a binary variable according to the traffic value of λ_f^{NODE} for each node. Based on this indicator, the traffic mean latency for IoT nodes is given as the value corresponding to the indicator in the lookup table.

4.3.4 Energy efficient - Low latency service embedding

In this scenario, we consider a multi-objective MILP model to optimise the service embedding in IoT networks to achieve a trade-off between minimising the power consumption and minimising the traffic mean latency. The objective function is given as:

$$\text{Objective: } \textbf{\textit{minimise}}\ \alpha.\mathrm{TL} + \beta.\mathrm{TNP} + \gamma.\mathrm{TPP} \qquad (4.7)$$

where α , β and γ are weight factors with the following units 1/ms, 1/mW, 1/mW respectively used to emphasise the importance of the different components of the objective function.

4.3.5 Framework Constraints

The framework performs the embedding operation through two parts as follows:

4.3.5.1 Embedding of virtual nodes

$$\sum_{c \in P} I_{iac}^{NE} = 1 \qquad (4.8)$$

$$\forall\, i \in B\,, \qquad \forall\, a \in V$$

$$\sum_{a \in V} I_{iac}^{NE} \;\leq 1 \tag{4.9}$$

$$\forall\, i \in B\,, \forall c \in\; P$$

Constraint (4.8) ensures that each virtual node in a BP is embedded in a single IoT node only. Constraint (4.9) states that each IoT node is not allowed to host more than one virtual node in each BP. This is considered as a coexistence constraint that is not used in all scenarios such as controller node virtualisation.

$$\sum_{i \in B} \sum_{a \in V} I_{iac}^{NE} \quad \geq I_c^{PM} \tag{4.10}$$

$$\forall\, c \in P$$

$$\sum_{i \in B} \sum_{a \in V} I_{iac}^{NE} \;\leq I_c^{PM} \cdot M \tag{4.11}$$

$$\forall\, c \in P$$

Constraints (4.10) and (4.11) build (include / add) a processing module in IoT node c if that node is chosen for embedding at least one virtual node a in BP i or more, where M is a large enough unitless number to ensure that I_c^{PMI} = 1 when $\sum_{i \in B} \sum_{a \in V} P_{iac}^{NE}$ is greater than zero.

$$\sum_{i \in B} \sum_{a \in V} V_{ia}^{MCU} \cdot\; I_{iac}^{NE} \;\leq\; P_c^{MCU} \tag{4.12}$$

$$\forall c \in P$$

$$\sum_{i \in B} \sum_{a \in L} V_{ia}^{RAM} \cdot\; I_{iac}^{NE} \;\leq\; P_c^{RAM} \tag{4.13}$$

$$\forall c \in P$$

Constraints (4.12) and (4.13) represent the processing and memory capacity constraints, respectively. They ensure that the embedded processing and memory workloads in an IoT node do not exceed the MCU and memory capacities, respectively.

$$I_{iac}^{NE} \cdot V_{ian}^{FUNC} = I_{iacn}^{F} \tag{4.14}$$

$$\forall i \in B,\ \forall a \in L,\ \forall c \in P,\ \forall n \in F$$

$$P_{cn}^{FUNC} >= I_{iacn}^{F} \tag{4.15}$$

$$\forall i \in B,\ \forall a \in L,\ \forall c \in P,\ \forall n \in F$$

Constraints (4.14) and (4.15) ensure that the required function of each virtual node in BP is provided by its hosting IoT node.

$$I_{iac}^{NE} \cdot V_{iaz}^{ZONE} = I_{iacz}^{Z} \tag{4.16}$$

$$\forall i \in B, \forall a \in V, \forall c \in P, \forall z \in Z$$

$$P_{cz}^{ZONE} \geq I_{iacz}^{Z} \tag{4.17}$$

$$\forall i \in B, \forall a \in V, \forall c \in P, \forall z \in Z$$

Constraints (4.16) and (4.17) ensure that the required zone of each virtual node in a BP is matched by the zone of the hosting IoT node.

4.3.5.2 Embedding of virtual links

$$I_{iac}^{NE} + I_{ibd}^{NE} = X_{iabcd}^{LE} + 2 \cdot I_{iabcd}^{LE} \tag{4.18}$$

$$\forall\, i \in B, \forall a \in V, \forall b \in VN_{ia} : a \neq b, \forall\, c,d \in P: c \neq d$$

Constraint (4.18) ensures that neighbouring virtual nodes a and b of i in B are also connected in the embedding IoT nodes c and d. We achieve this by introducing a binary variable P_{iabcd}^{LE} which is only equal to 1 if I_{iac}^{NE} and I_{ibd}^{NE} are exclusively equal to 1 otherwise it is zero, W_{iabcd}^{LE} is an auxiliary variable.

$$\sum_{i \in B} \sum_{a \in L} \sum_{b \in LNB_{i\,a}} I_{iabcd}^{LE} \cdot V_{iab}^{TRFIC} = R_{cd}^{TRFP} \tag{4.19}$$

$$c,d \in P: c \neq d$$

Constraint (4.19) generates the path's traffic matrix resulting from embedding the virtual nodes a and b into the IoT nodes c and d.

$$\sum_{f \in PN_e} R_{cdef}^{ROUTE} - \sum_{f \in PN_e} R_{cdfe}^{ROUTE} \begin{cases} R_{cd}^{TRFP} & if\, e = c \\ -R_{cd}^{TRFP} & if\ e = d \\ 0 & otherwise \end{cases} \tag{4.20}$$

$\forall\, c, d, e \in P: c \neq d \; and \; e \neq f$

Constraint (4.20) represents the flow conservation constraint for the traffic flows in the IoT network.

$$\sum_{c \in P} \sum_{d \in P} R_{cdef}^{ROUTE} = R_{ef}^{TRFL} \tag{4.21}$$

$\forall\, e \in P\,, \forall\, f \in PN_e$

Constraint (4.21) estimates link's traffic between the neighbouring IoT nodes *e* and *d.*

$$\sum_{f \in PN_e} R_{ef}^{TRFL} \leq P_e^{CAPT} \tag{4.22}$$

$\forall\, e \in P$

Constraint (4.22) states that the total traffic flows of the IoT node *e* should not exceed the node capacity i.e. 250 kbps.

$$R_{cdef}^{ROUTE} \geq I_{cdef}^{R} \tag{4.23}$$

$\forall\, c, d, e \in P, \forall f \in PN_e : c \neq d\,, e \neq f$

$$R_{cdef}^{ROUTE} \leq I_{cdef}^{R} \cdot M \tag{4.24}$$

$\forall\, c, d, e \in P, \forall f \in PN_e : c \neq d\,, e \neq f$

The constraints (4.23) and (4.24) build a path between the embedding IoT nodes *c* and *d* through the neighbouring IoT nodes *e* and *f*, where $I_{cdef}^{RI} = 1$ if there is a traffic path between the IoT nodes *c* and *d* that passes through the neighbouring IoT nodes *e* and *f*, where *M* is a large enough unitless number which ensure that I_{cdef}^{R} = 1 when R_{cdef}^{ROUTE} is greater than zero.

$$\sum_{f \in PN_e} I_{cdef}^{R} \leq 1 \tag{4.25}$$

$\forall c \in P, \forall d \in P, \forall e \in P$

Constraint (4.25) ensures that traffic splitting is prevented for each path between the embedding IoT nodes *c* and *d*, such that the maximum number of physical links between neighbouring IoT nodes e and f is one.

$$\sum_{c\in P}\sum_{d\in P}\sum_{f\in PNB_e} I^R_{cdef} \geq I^{TM}_e \qquad (4.26)$$

$\forall e \in P$

$$\sum_{c\in P}\sum_{d\in P}\sum_{f\in PNB_e} I^R_{cdef} \leq I^{TM}_e \cdot M \qquad (4.27)$$

$\forall e \in P$

Constraints (426) and (4.27) build a network module in IoT node *e* if that IoT node is chosen to send/receive traffic at least for one link or more, where *M* is a large enough unitless number to ensure that I^{TMI}_e=1 when $\sum_{c\in P}\sum_{d\in P}\sum_{f\in PNB_e} I^R_{cdef}$ is greater than zero.

$$\sum_{e\in PN_f} R^{TRFL}_{ef} = R^{TRFN}_f \qquad (4.28)$$

$\forall f \in P : e \neq f$

Constraint (4.28) estimates the arrival traffic for each IoT node.

$$\sum_{j\in J} LI^{LMBDA}_{f\ j} \cdot j = R^{TRFN}_f \qquad (4.29)$$

$\forall f \in P : e \neq f$

Constraint (4.29) is an arrival rate indicator of arrival rate *j* for each IoT node *f*

$$\sum_{j\in J} LI^{LMBDA}_{f\ j} \leq 1 \qquad (4.30)$$

$\forall f \in P$

Constraint (4.30) ensures that each IoT node has no more than one arrival rate indicator.

$$\sum_{j\in J} W^{LIMDA}_j \cdot LI^{LMBDA}_{f\ j} = W^{NODE}_f \qquad (4.31)$$

$\forall f \in P$

Constrain (4.31) estimates the mean traffic latency for each IoT (f).

The MILP optimisation model was solved using CPLEX running on personal computer with processor core i5 -3.2 GHz and 16 GB RAM and on the university Polaris servers of 24 cores and 128GB.

4.4 Results and evaluations

To evaluate the performance of the proposed model and heuristic, we consider a smart building scheme (for example in an enterprise campus) where the physical layer is composed of 30 IoT nodes connected by 89 bidirectional wireless links. These IoT nodes are distributed across an area 500 m x 500 m and can carry various functions with the following assumptions:

- There is a set of 9 distinct functions, 4 sensing functions, one control function and 4 actuating functions. Each IoT node can provide 2 sensing function, 2 actuating functions, and one controlling function (present only in one type of processor). The virtual node of each BP requests one function only.
- There is a set of five geographical zones that represent the sub-sections of the smart building (e.g. departments or sections in the enterprise campus). Each zone is equipped with six IoT nodes. All the functions and processor types exist in each zone. The virtual node requests an embedding location in one of these five zones.
- The IoT nodes processing capability is uniformly distributed among five processing capacities (8, 16, 16, 25, 25, 48 MHz) representing microcontrollers as shown in Table 4-1. Each virtual node has a specific processing demand that varies between 4 and 30 MHz.
- Each IoT node contains wireless transceiver modules [211]. The network modules used are low cost, low power, and are compatible with the ZigBee protocol stack for IoT networks [209]. The traffic demands of the virtual links vary from 50 to 200 packets per second with a packet size of 1 kb.
- We study the embedding of 12 BPs arriving sequentially, two at a time. Each BP has three virtual nodes (sensor, controller and actuator) connected sequentially. The sensor is connected to the controller and the controller is connected to the actuator. The sensor virtual node requests a specific sensing function, the control virtual node requires processing capacity and the actuator

virtual node requests a specific actuating function. The sensor and actuator virtual nodes of a BP need to be embedded in a specific zone while the controller virtual node can be embedded into any geographical zone.

Table 4-1: Processing modules power specifications and power consumption in active mode[212]

MCU Type	MCU CLK	RAM	Idle Power	Max. Power
MSP430F1	8 MHz	64 kB	1 mW	8 mW
MSP430FR5	16 MHz	64 kB	1 mW	14 mW
MSP430FR6	16 MHz	128 kB	1 mW	20 mW
MSP430F5	25 MHz	512 kB	1 mW	14 mW
MSP432P4	48 MHz	256 kB	1 mW	16 mW

We evaluate the power consumption and traffic mean latency resulting from embedding the BPs using the MILP model considering the three objective functions discussed in Section 4.3.

4.4.1 Energy efficient service embedding

In this section, we evaluate the results of embedding BPs in terms of power consumption and traffic mean latency under three scenarios with objective function used in (4.1). In the first scenario, referred to as energy-latency unaware service embedding (ELUSE), BPs are embedded in physical nodes and links that satisfy their requirements with no consideration of a certain objective function.

In the second and third scenarios, the objective is to minimise the total power consumption. However, in the second scenario, referred to as re-provisioning, each time a new BPs arrives, previously embedded BPs are re-embedded while in the third scenario, referred to as sequential embedding, arriving BPs are embedded without interrupting the existing BPs. We also study the coexistence constraints of the embedding and their effects on the results of the energy efficient service embedding.

4.4.1.1 Service embedding on same geographical zone

In this subsection, we considered that the sensor and actuator nodes of a BP need to be embedded in the same specific geographical zone. We also study embedding BPs with and without coexistence constraints. Under coexistence constrains, the virtual nodes of the BP cannot coexist in the same IoT node. The goal here is to improve the resilience of the BPs under single node failure.

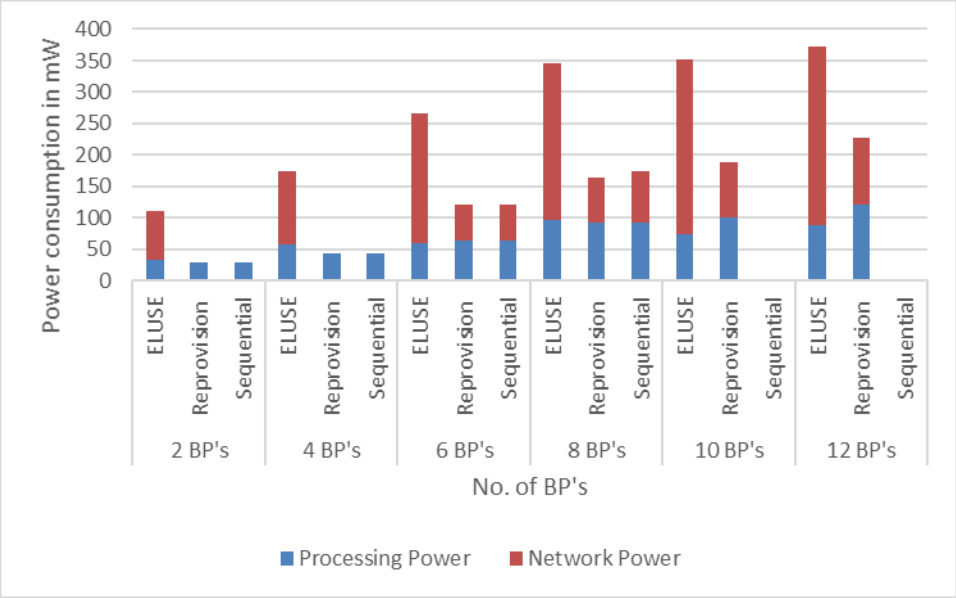


Figure 4-5: Power consumption of energy efficient service embedding in same zone without coexistence constraint.

Figure 4-5 shows the total power consumption of embedding BPs in which the sensing and actuating nodes are to be embedded in the same zone. The results show that the energy efficient re-provisioning embedding scenario resulted in saving an average of 63% of the power consumption compared to the ELUSE scenario. Under energy efficient embedding, fewer IoT nodes and links are activated to embed BPs compared to embedding under the ELUSE scenario. As no coexistence constraints apply, all the virtual nodes of a BP can be embedded in a single IoT node confining the virtual links traffic within this node and reducing the number of activated IoT nodes. The saving achieved by the energy efficient embedding decreases to 58% under the sequential scenario as the sequential approach builds on existing embedding decisions that become suboptimal with the arrival of new BPs. The optimal use of resources under the re-provisioning scenario resources resulted in successfully embedding 12 BPs while only 8 BPs were successfully embedded under sequential embedding, that refers to the re-provisional scenario allocates the

virtual nodes in to the optimal physical resources from the whole network, while the sequential scenario allocates the incoming virtual nodes in to the remaining physical resources which is not satisfy the higher virtual demands such as in more than 8 BP's.

Note that the power savings decrease as the number of embedded BPs increases. This is because the higher the load on the network the fewer the possible embedding solutions therefore narrowing the gap between energy efficient embedding and ELUSE.

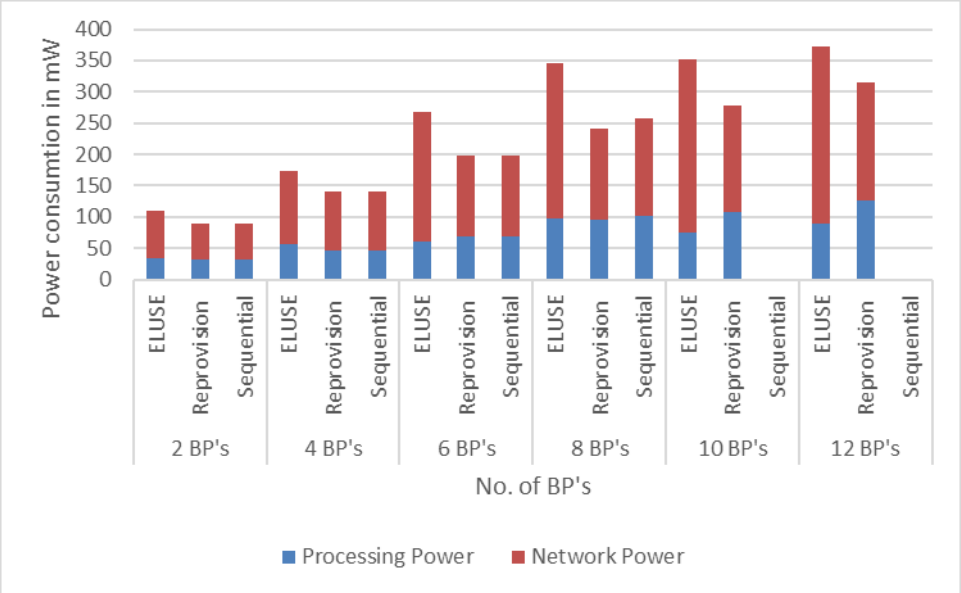


Figure 4-6: Power consumption of energy efficient service embedding in the same zone with coexistence constraint.

Figure 4-6 shows the power consumption of embedding BPs in the same zone under coexistence constraints. The coexistence constraints reduce the power savings achieved by the energy efficient embedding scenarios to 36% and 29% for re-provisioning and sequential embedding, respectively. This reduction in power savings is due to the need to activate more IoT nodes to meet the coexistence requirements and the traffic between these nodes.

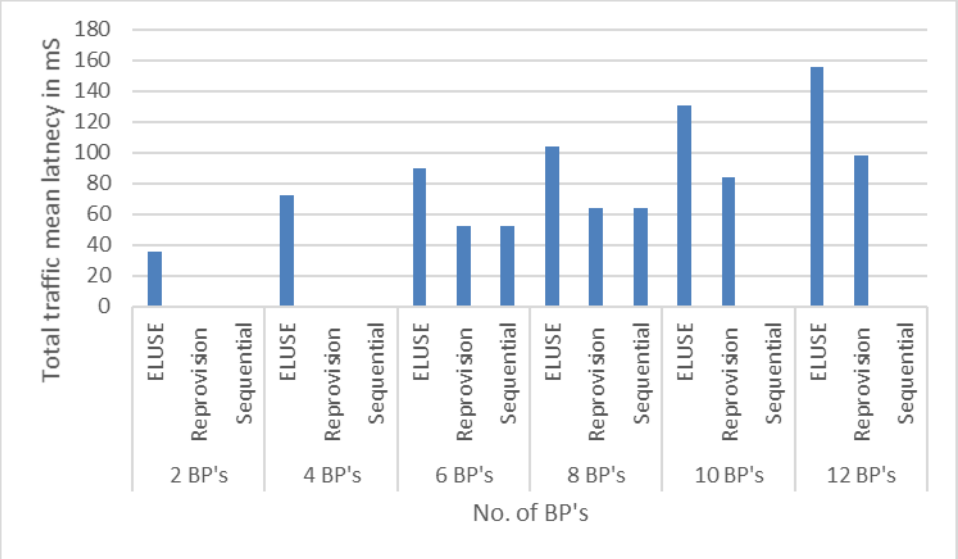


Figure 4-7: Average traffic mean latency of energy efficient service embedding in same zone without coexistence constraint.

The results in Figure 4-7display the average traffic mean latency resulting from embedding BPs without coexistence constraint. The re-provisioning embedding and the sequential embedding have reduced the average traffic mean latency by 62% and 60% respectively compared with ELUSE scenario. This is because energy efficient embedding selects routes of minimum hops and consequently lower traffic mean latency compared to random routing in ELUSE. However, energy efficient embedding does not produce the minimum traffic mean latency (as we will see in Section 4.4.2) as energy efficient embedding tries to highly utilise the activated IoT nodes resulting in high traffic mean latency in these nodes.

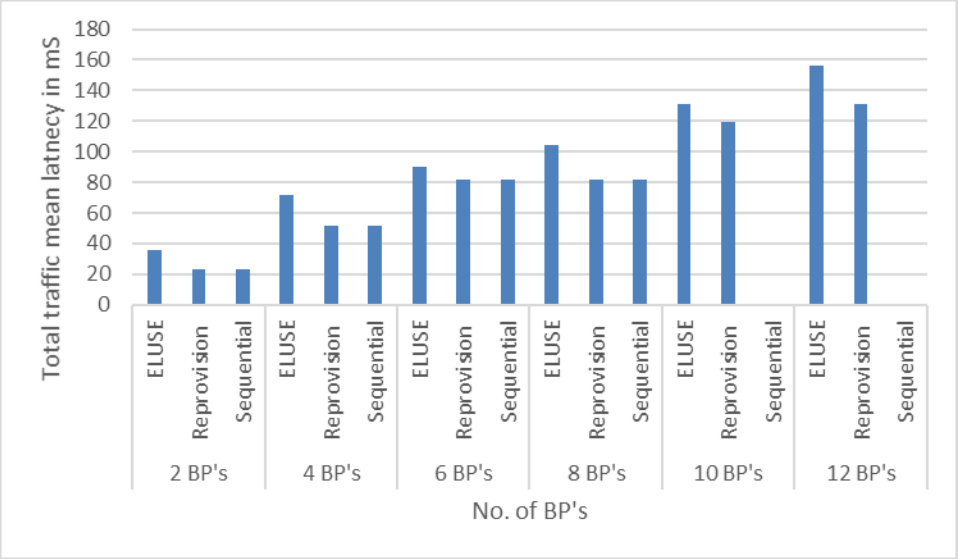


Figure 4-8: Average traffic mean latency of energy efficient service embedding in same zone with coexistence constraint

Similar trends to those in Figure 4-7 are observed in Figure 4-8 for the average traffic mean latency resulting from embedding with coexistence constraints, the results display that the re-provisional embedding and the sequential embedding have reduced the average traffic mean latency by 27% compared with the ELUSE scenario. Comparing Fig. 8 and Fig.7 shows that embedding BP on the same zone with coexistence constraint results in higher traffic mean latency compared to embedding without coexistence constraint. This is because without the coexistence constraint, the traffic of a BP can experience no traffic latency by embedding all the virtual nodes of the BP in a single IoT node.

4.4.1.2 Service embedding across geographical zone

The previous results display the power consumption and mean latency of embedding BPs where the sensor and actuator nodes need to be embedded in the same geographical zone. In this section we examine embedding BPs that require the sensor and actuator nodes to be embedded in distinct geographical zones. We study also the performance with and without coexistence constraints on the controller node. Under coexistence constrains, the controller cannot coexist in the same IoT node with the sensor or actuator node.

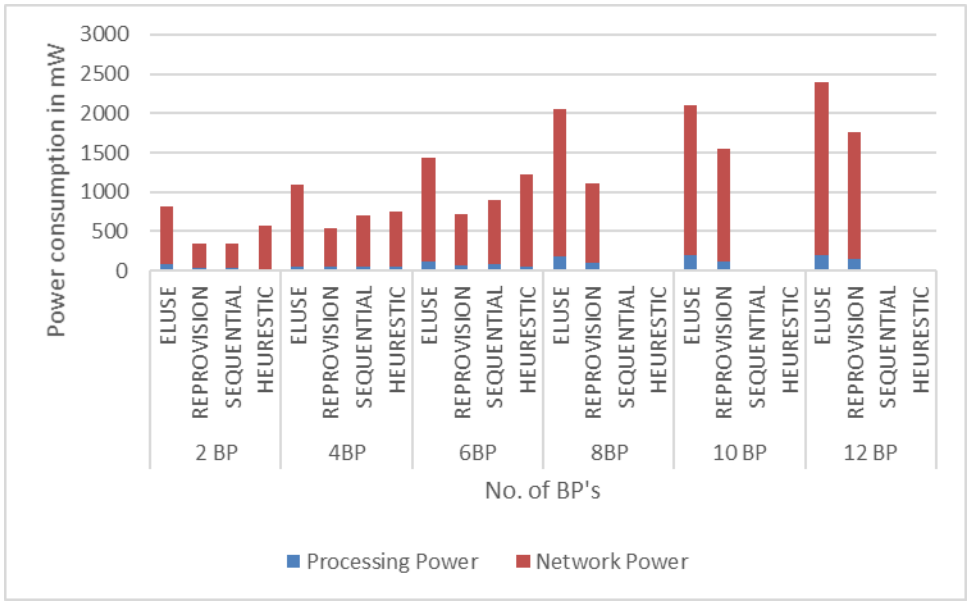


Figure 4-9: Power consumption of energy efficient service embedding across different zones without coexistence constraint.

Figure 4-9 displays the power consumption of embedding BPs across different geographical zones without coexistence constraint. The power savings achieved by energy efficient embedding under the re-provisioning scenario and the sequential scenario when embedding across different zones are lower than those achieved for same zone embedding in Fig. 6. This is because energy efficient embedding in the distinct zones cannot select to embed the sensor and actuator in the same node although coexistence constraints do not apply. The power savings achieved by the energy efficient embedding scenarios are 42% and 22% for re-provisioning and sequential scenarios, respectively.

The less efficient use of resources in embedding across zones reduces the number of BPs that can be embedded under the sequential scenario to 6 BPs, while the re-provisioning embedding still succeeds to embed all the 12 BPs.

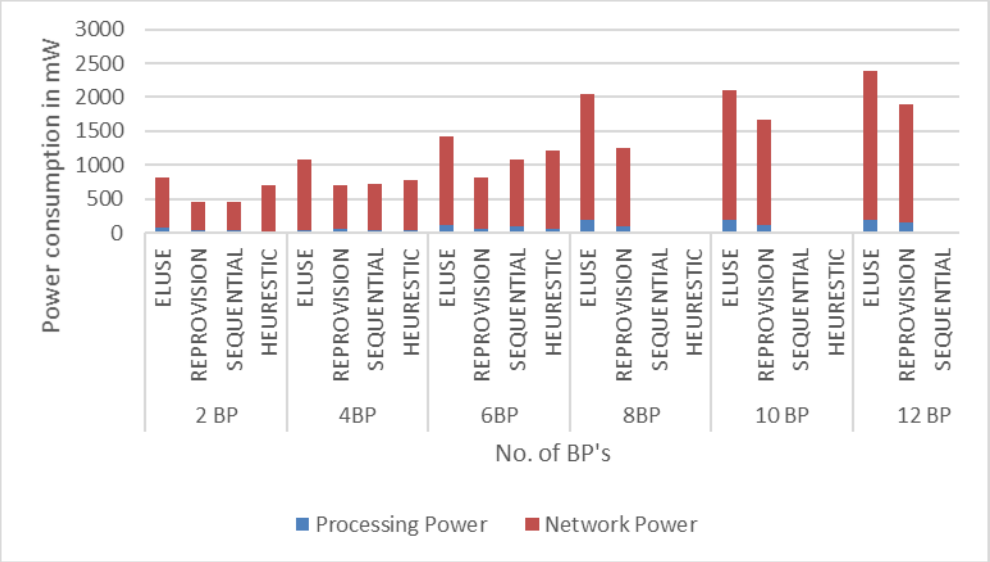


Figure 4-10: Power consumption of energy efficient service embedding across different zones with coexistence constraint.

Figure 4-10 displays the power consumption of embedding BP's into the physical IoT network with the coexistence constraint. The power savings achieved by the energy efficient embedding scenarios are reduced to 34% and 17% for re-provisioning and sequential cases, respectively. This reduction is due to embedding of virtual nodes of a BP in different IoT nodes as explained above.

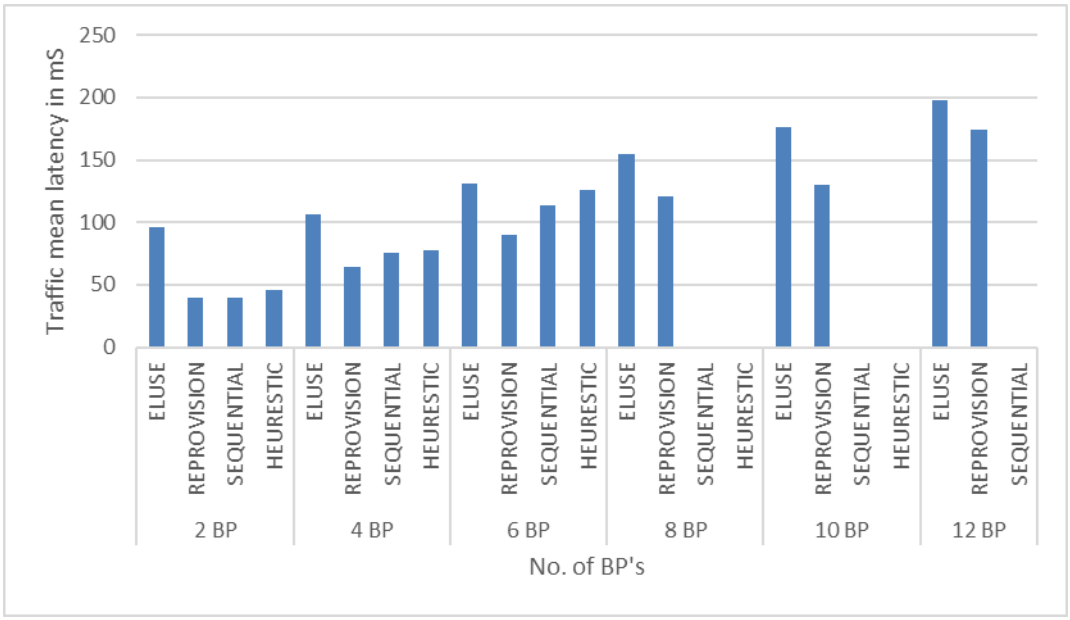


Figure 4-11: Average latency of energy efficient service embedding across different zones without coexistence constraint.

Figure 4-11 displays the traffic mean latency resulting from embedding BPs across distinct zones without coexistence constraints. The re-provisioning and sequential embedding have reduced the average traffic mean latency by 32% and 15% compared with ELUSE scenario.

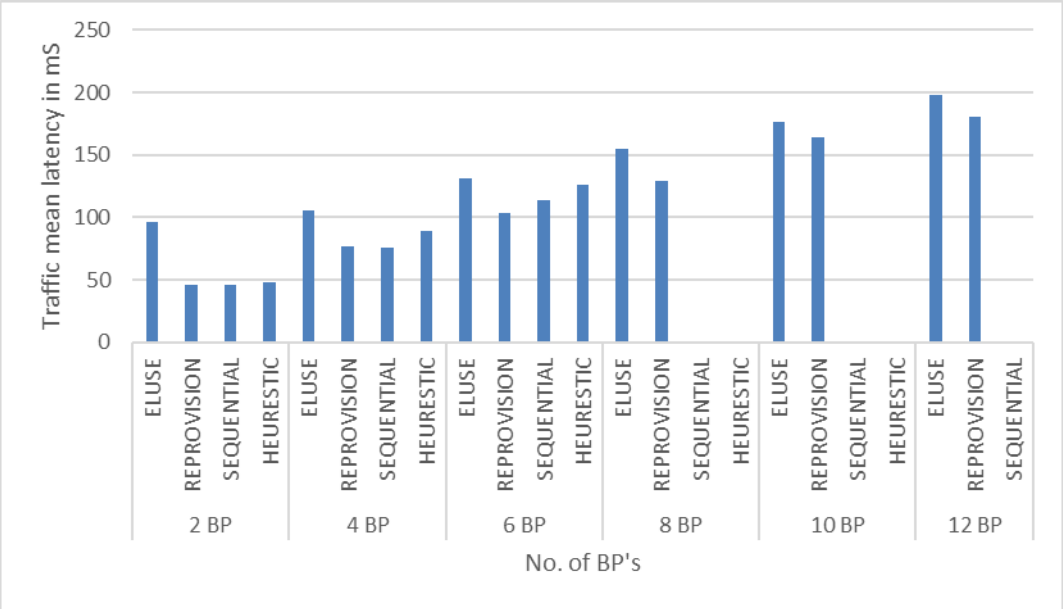


Figure 4-12: Average latency of energy efficient service embedding across different zones with coexistence constraint.

Figure 4-12 displays the traffic mean latency resulting from embedding BPs across distinct zones without coexistence constraints. The re-provisioning and sequential embedding have reduced the average traffic mean latency to 22% and 13% compared with ELUSE scenario.

4.4.2 Low latency service embedding in IoT networks

In this subsection, we evaluate the low traffic mean latency embedding of BPs across different zones with and without the coexistence constraint in terms of traffic latency and power consumption with objective function used in (4.4).

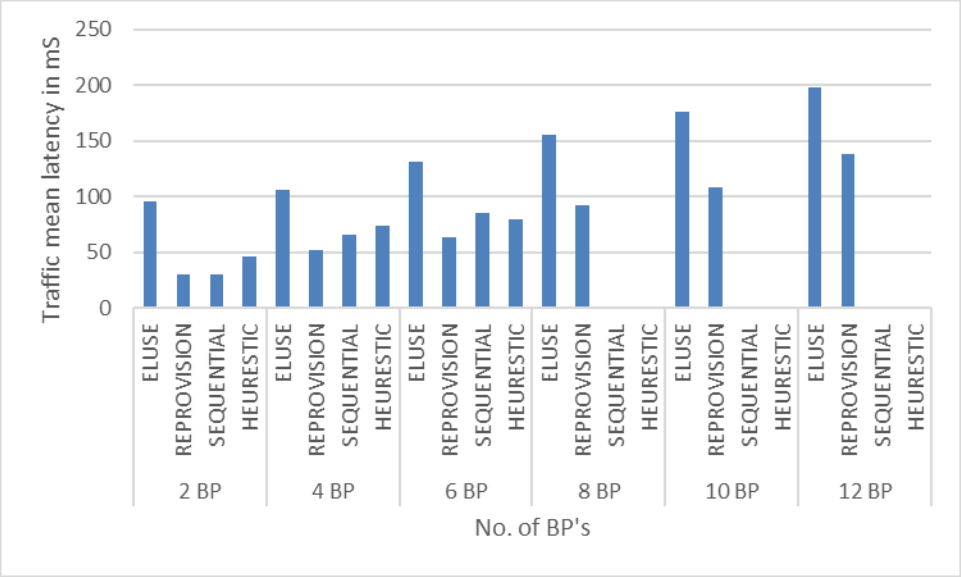


Figure 4-13: Average traffic mean latency of low latency service embedding across different zones without coexistence constraint.

Figure 4-13 shows that the re-provisioning low latency embedding resulted in reducing the traffic latency by an average of 47% compared to the ELUSE scenario. The low latency embedding model optimises the selection of IoT nodes and distributes the traffic so the arrival rate at nodes and consequently the traffic latency is minimised. Under energy efficient embedding, fewer IoT nodes and links are activated to embed BPs compared to embedding under the ELUSE scenario.

The traffic latency reduction achieved by the energy efficient embedding decreases to 20% under the sequential scenario as the sequential approach builds on existing embedding decisions as explained in Section 4.3.1. The optimal use of resources under the re-provisioning scenario resulted in successfully embedding 12 BPs while only 6 BPs were successfully under the sequential embedding.

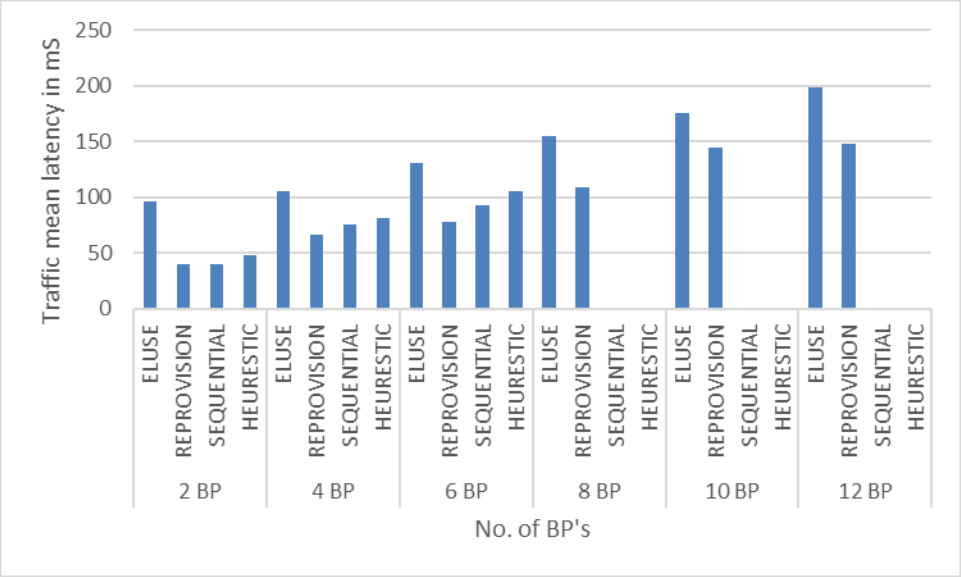


Figure 4-14: Average traffic mean latency of low latency service embedding across different zones with coexistence constraint.

Figure 4-14 displays the traffic mean latency of low latency BPs embedding across different zones with coexistence constraint. Adding the coexistence constraint reduced the traffic latency achieved by the re-provisioning and sequential embedding to 34% and 19%, respectively compared to the ELUSE scenario as more traffic traverses the network due to the fact that multiple virtual nodes of the same BP cannot coexist on the same IoT node.

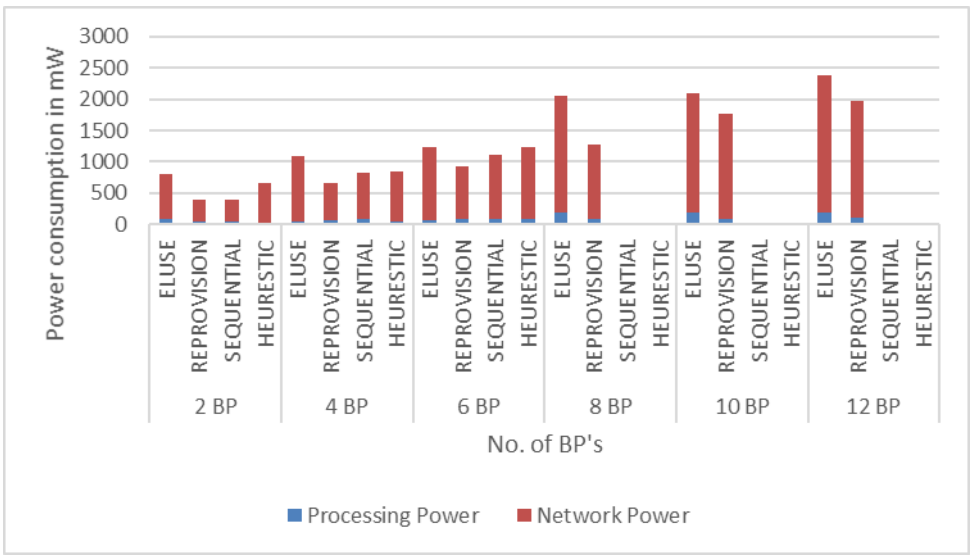


Figure 4-15: Power consumption of low latency service embedding across distinct zones without coexistence constraint.

The results in Figure 4-15 show the power consumption resulting from low latency embedding across distinct zones without coexistence constraint. Distributing the traffic to reduce the delay increased the power consumption by 28% compared to the energy efficient re-provisioning embedding in Fig. 9 as more nodes are activated. However, compared to the ELUSE scenario the power consumption is reduced by 18% and 10% under low latency re-provisioning and low latency sequential embedding, respectively.

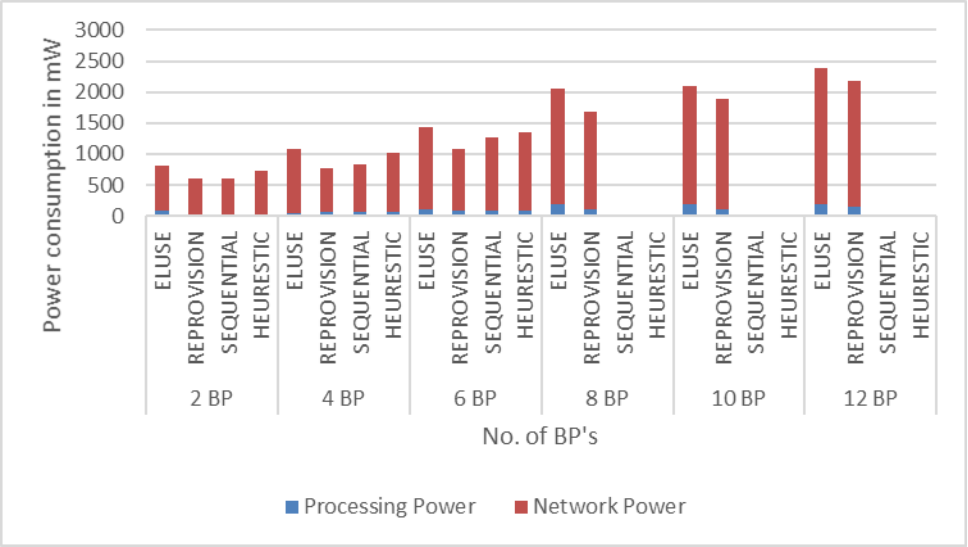


Figure 4-16: Power consumption of low latency service embedding across distinct zones with coexistence constraint.

Under the coexistence constraint in Figure 4-16, the increase in power consumption resulting from low latency embedding compared to the energy efficient embedding increased the power consumption by 20% compared to the energy efficient re-provisioning embedding in Fig. 10. However, compared to the ELUSE scenario the power consumption is reduced by 14% under low latency re-provisioning and sequential embedding.

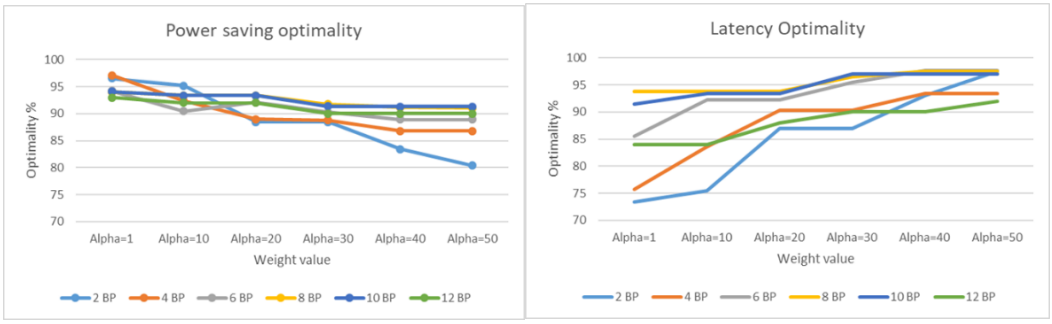
4.4.3 Energy efficient-Low latency service embedding in IoT networks

Minimum power consumption is achieved by consolidating the embedding of virtual nodes in as few as possible energy efficient IoT nodes. On the other hand, minimum traffic mean latency is achieved by distributing the traffic into multiple paths to reduce the arrival rate at the individual IoT nodes. As explained in Section

4.3.4, the trade-off between minimising the power consumption and minimising the traffic mean latency is achieved through a multi-objective MILP model.

We define a metric referred to as "embedding optimality" to compare the performance of the multi-objective embedding to single objective embedding. The embedding optimality is defined as follows:

$$Optimality^{QoS} = \frac{Optimal^{QoS}_{Multi-objective}}{Optimal^{QoS}_{Single-objective}} \quad (4.32)$$



(A) **(B)**

Figure 4-17: Optimality of (a) power saving and (b) traffic mean latency of embedding in distinct zones with coexistence constraint.

Figure 4-17 displays the power saving (Figure 4-17.A) and traffic mean latency (Figure 4-17.B) average optimality of energy efficient–low latency service embedding scenario across distinct zones with coexistence constraint under $\alpha = 30$, $\beta = 1$ and $\gamma = 1$ in the multi-objective function (equation (4.7)). Note that the numerical value of power consumption and traffic latency are comparable therefore the weight α is used to prioritise traffic latency, while the other two weights in equation (4.7) are set to one. We obtain equal optimality for power savings and mean traffic latency of 91% at α=30, i.e. this is the weight needed to achieve the trade-off.

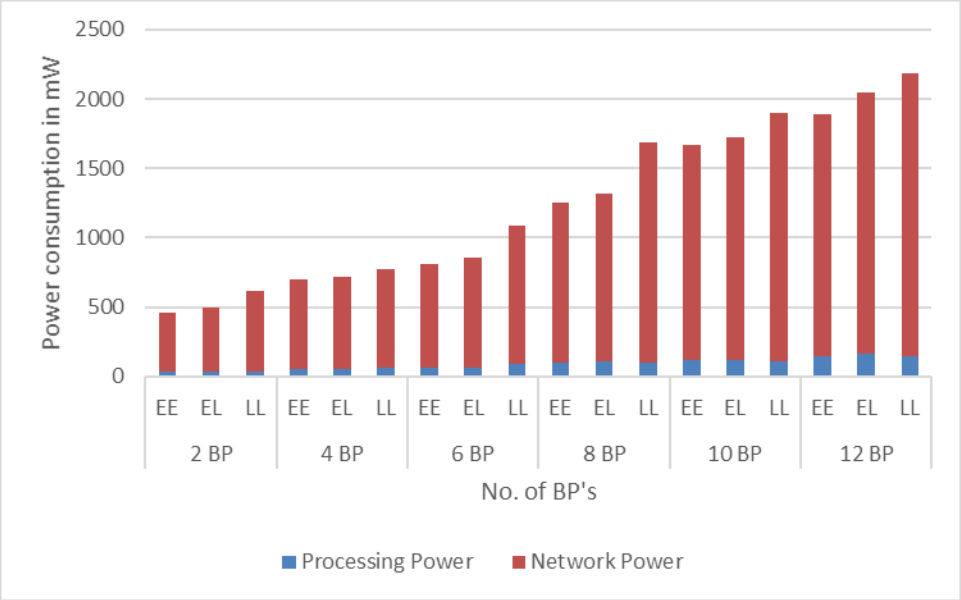


Figure 4-18: Power consumption of embedding in distinct zones with coexistence constraint.

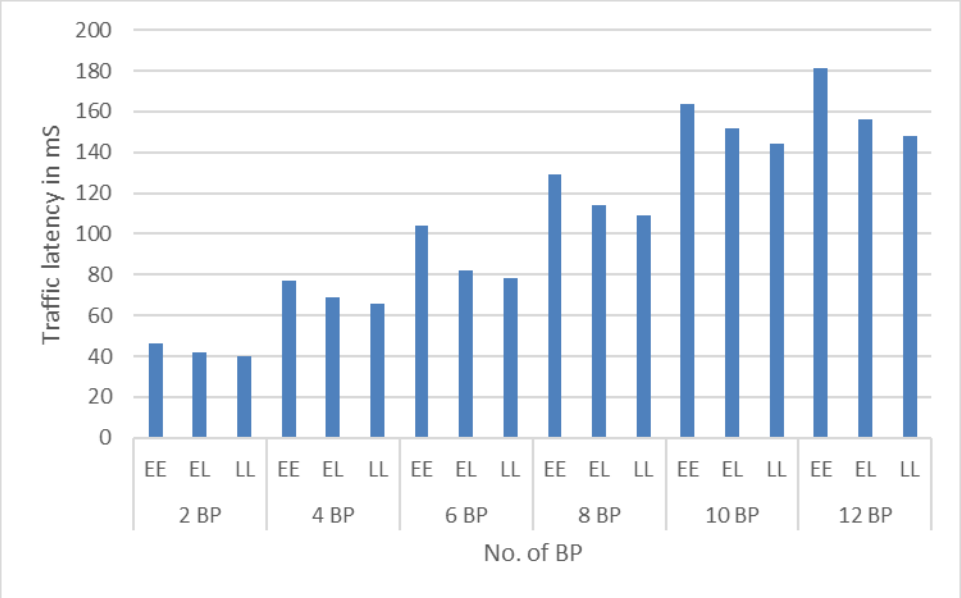


Figure 4-19: Average traffic mean latency of embedding in distinct zones with coexistence constraint.

Figure 4-18 and Figure 4-19 compare the power consumption and delay, respectively of the energy efficient–low latency service embedding scenario with $\alpha = 30$ to those of the energy efficient service embedding and low latency service embedding scenarios. Note that the low latency scenario increases the power consumption by 20% compared to the energy efficient scenario (Figure 4-18) and

the energy efficient scenario increases the traffic mean delay by 22% compared to the low latency scenario (Figure 4-19).

4.5 Real time service embedding heuristics

We have developed service embedding heuristics for two purposes:

i) For real time service embedding due to the high complexity of the MILP model

ii) To validate the MILP results.

The two heuristics are developed; real time energy efficient service embedding (RESE) heuristic and real time low latency service embedding (RLSE) heuristic. These heuristics can be applied for a general embedding scenario.

4.5.1 Real time energy efficient service embedding heuristic

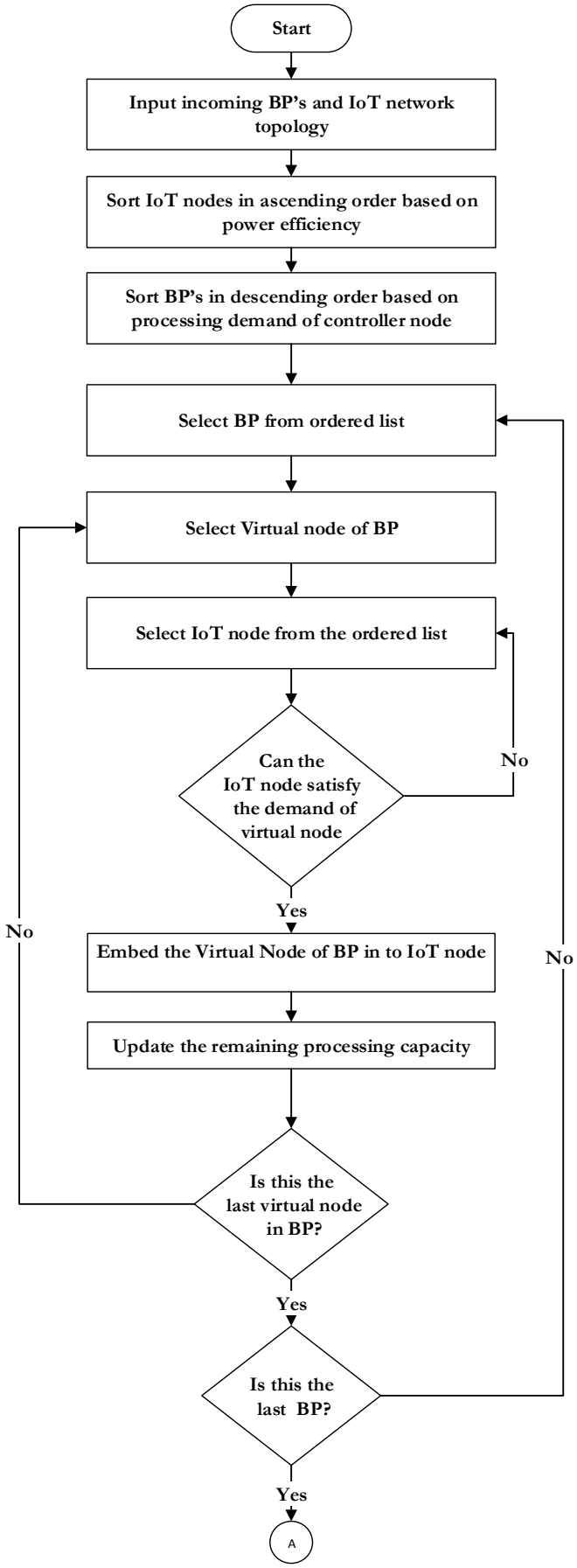
The flowchart of the RESE heuristic is shown in Figure 4-20. The input to the heuristic is the IoT network topology and the BPs. The heuristic starts by sorting the IoT nodes according to the processing power efficiency in descending order and the BPs according to the processing demand of the controller node in ascending order. The heuristic picks a BP from the ordered list and embeds its nodes one by one considering the IoT node with the highest energy efficiency that satisfies the embedding requirements in terms of function, zone and coexistence. By doing so the heuristic tries to consolidate virtual nodes into the most energy efficient IoT node that meets its demand before activating another IoT node. The available processing capacity of the IoT nodes is updated after the embedding of a virtual node and another virtual node of the BP is selected to be embedded. After embedding all the virtual nodes of a BP, the traffic between the virtual nodes is routed based on finding shortest path routing [213]. This process is repeated for all BPs and the total power consumption (IoT nodes and network) and traffic mean latency resulting from embedding all the BPs are calculated.

Figure 4-9 to Figure 4-12 show that the performance of the RESE heuristic approaches that of the sequential energy efficient MILP model for embedding across

different zones. Table 4-2 summarises the average performance gap between the RESE heuristic and the sequential model.

Table 4-2: Power consumption gap between the RLSE heuristic and the sequential model.

	2 BP's		**4 BP's**		**6 BP's**	
	Sequential MILP	**Real time Heuristic**	**Sequential MILP**	**Real time Heuristic**	**Sequential MILP**	**Real time Heuristic**
Processing Power	35	23	48	44	96	56
Network Power	420	684	672	736	987	1149


Start
Input incoming BP's and IoT network topology
Sort IoT nodes in ascending order based on power efficiency
Sort BP's in descending order based on processing demand of controller node
Select BP from ordered list
Select Virtual node of BP
Select IoT node from the ordered list
Can the IoT node satisfy the demand of virtual node
No
Yes
No
Embed the Virtual Node of BP in to IoT node
Update the remaining processing capacity
Is this the last virtual node in BP?
No
Yes
Is this the last BP?
Yes
A

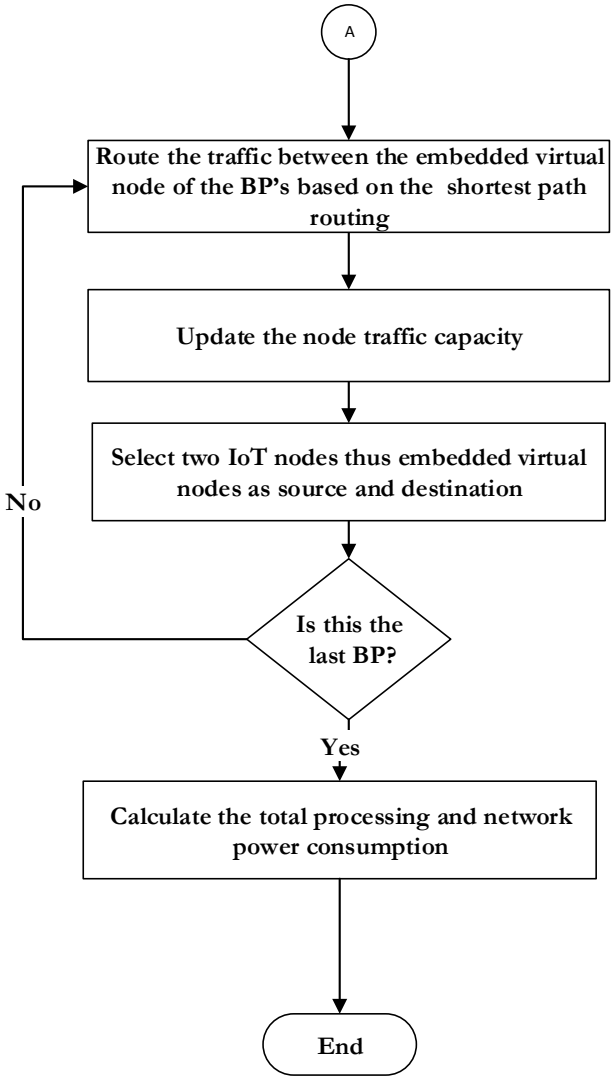


Figure 4-20: RESE Heuristic Flowchart.

4.5.2 Real time low latency service embedding heuristic

The RLSE heuristic reduces the traffic mean latency by setting a threshold on the node transmission capacity utilisation. When routing the traffic between virtual nodes of a BP, the heuristic does not exceed this threshold which grantees distributing the traffic over multiple links. The flowchart of the RLSE heuristic is given in Fig.4-20. The threshold is set to 60% of the maximum node capacity. Different thresholds were examined, and this threshold value was identified as the maximum threshold before the latency per node starts increasing fast.

Figure 4-13- Figure 4-16 show that the performance of the RESE heuristic approaches that of the sequential low latency MILP model for embedding across

different zones. Table 4-3 summarises the average performance gap between the RLSE heuristic and the sequential model.

Table 4-3: Traffic mean latency gap between the RLSE heuristic and the sequential model

	2 BP's		**4 BP's**		**6 BP's**	
	Sequential MILP	**Real time Heuristic**	**Sequential MILP**	**Real time Heuristic**	**Sequential MILP**	**Real time Heuristic**
Traffic mean Latency	40	48	76	81	93	106

4.6 Summary

This chapter has investigated the power consumption and traffic mean latency of service embedding in the IoT network for a smart building setting and has introduced a framework for their minimisation. The services to be embedded are represented by a virtual topology (virtual nodes and links) following a business processes workflow dictated by the SOA paradigm. We developed a MILP framework and a real-time heuristic to optimise the selection of IoT nodes to embed the virtual nodes; and to route the traffic between virtual nodes considering three different objective functions: (i) minimising the total power consumption, (ii) minimising traffic mean latency, (iii) minimising both total power consumption and traffic mean latency in multi-objective manner.

We considered embedding BPs where all the sensor and actuator nodes exist in the same geographical zone and also considered embedding across different zones. We also studied embedding with and without constraints on the coexistence of virtual nodes in the same IoT node.

We used the MILP model to optimise the embedding in two scenarios: (i) re-provisioning scenario where each time a new BPs arrives, previously embedded BPs are re-embedded, (ii) sequential embedding where arriving BPs are embedded without interrupting the existing BPs.

In the energy efficient service embedding scenario, the re-provisioning scenario produces higher average power saving compared with the sequential embedding scenario. In the low latency service embedding scenario, re-provisional embedding reduced the average traffic mean latency compared with the sequential embedding scenario. The multi-objective optimisation shows that it is possible to optimise the embedding of BPs to achieve high optimality of 91% for both power savings and traffic latency.

Chapter 5 Energy-efficient Service Embedding in Smart Cities with Fog and Cloud processing and latency minimisation

5.1 Introduction

To reduce the cost, size and power consumption, the IoT devices have been designed with limited data processing, storage, and traffic capabilities. These limitations are considered as the main constraints and blockages in developing a complex service with high computation and traffic demands. To overcome this problem, there is a significant demand for integration between the IoT and systematic resources that complement the IoT components to satisfy the requirements of the services. The cloud and fog are the main contributors when it comes to coordinating and providing processing capability, data storage, and resource management. The integration paradigm of the IoT and cloud leads to the success of the IoT world in terms of service provision accomplished with high performance, reliability, and scalability.

This chapter introduces a generic MILP model that has been developed to minimize the power consumption due to both processing by hosting server/node, and the traffic flow through the wireless/optical network. The model selects the processing hosting server/node according to the job processing completion latency. We apply this model to simulate a smart city setting. We investigate the power consumption of services embedding a cloud and fog centric IoT paradigm and evaluate the impact of latency constraint on the total power consumption. The latency due to job processing completion is typically much higher than the latency due to link congestion. Also, the propagation latency considering the longest path the traffic will travel between the IoT nodes and the cloud is negligible compared to the job processing latency. Therefore, this chapter focuses on latency due to job processing completion. We formulate the problem of finding the optimal set of nodes and links to embed BPs in the network as a MILP model considering as an objective function the minimisation of network and processing power consumption, with latency constraints.

5.2 Proposed Architecture

In the smart city, there are distinct applications that employ the same resources in the monitoring and controlling system. An example of such an application is traffic and security monitoring, where these services refer to a number of different paradigms for enhancing the operation of autonomous cars, traffic lighting, and monitoring the security condition of the road in the smart city scheme; the majority of these paradigms depend on IoT nodes and are thus embedded with different types of sensors and actuators (i.e. cameras, street's display screens, traffic light controls). These devices need to continuously monitor and control several crucial factors, such as vehicle registration numbers, traffic congestion monitoring, and speed of vehicles, the existence of pedestrians around road crossings lines, climate situations, and securing public places from unauthorised intrusion.

These kinds of services require higher processing and traffic demands than the services studied in Chapter 4; the control system should make the decision and execute different operations with high performance and efficiency to deliver different services, i.e. traffic lights control, road monitoring, accident alarms and information distribution, in an intelligent manner. Consequently, there is a significant demand for integration between the IoT and other technologies (i.e. cloud and fog computing) [214], [215], [63], [216]. The integration paradigm of the IoT and cloud leads to the success of the IoT world in terms of service provision accomplished with high performance, reliability, and scalability [217]. The cloud features are provided with high elasticity and on-demand resources for efficient and scalable service provision [30]. Although cloud computing is an emerging technology which processes content for distributed environments, the cloud's property of centralised computing comes with a high traffic overhead, such as capacity and challenges linked to power consumption [218], [219]; hence, another paradigm called fog computing is being developed to meet these requirements [69].

Fog computing is considered a cloud-derived solution and is based on the distribution of computing resources and services nearer to the endpoints of the network edge. The computing resources of the fog are in the local network, such that there is no need to send data to the cloud for processing and storage. Fog computing cannot be considered a system independent from the cloud, because the related

processing components in the fog and cloud need to exchange important updates and remain synchronised [34] .

Besides the challenges of emerging computing technologies, there is another trend of development related to the network of a huge number of IoT nodes which generate a substantial amount of data (e.g. camera video streaming). There is a wide range of network technologies connecting IoT devices to the cloud/fog; these network technologies aggregate a huge amount of data traffic from vast individual sensor nodes with different traffic loads through the access network.

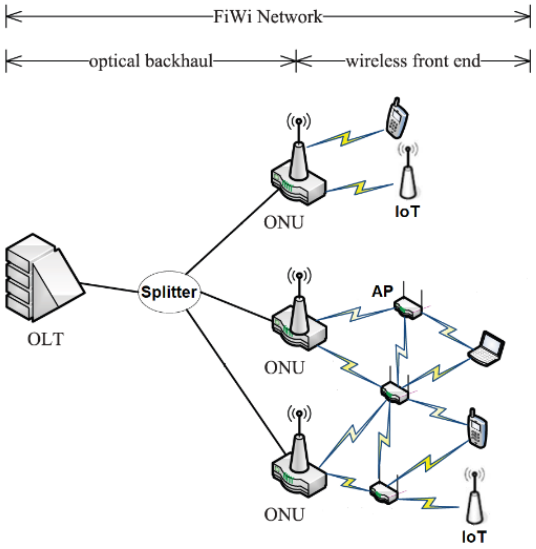


Figure 5-1: Schematic access network structure.

One of these energy-efficient paradigms is Fibre-Wireless (FiWi) networks [220]. As shown in Figure 5-1, the FiWi networks adopt optical networks like Passive Optical-fibre Networks (PONs) as their backhaul to provide high capacity and reliability [221], [222]; on the other side, the FiWi networks enhance the ubiquitous coverage, connectivity, and mobility of wireless networks and thus present the front end network like Wi-Fi. The structure of the FiWi networks can be summarised as shown in Fig. 1 by:

- Optical Backbone of the FiWi, where the PON provides high link capacity by exploiting multiplexing techniques over optical fibre networks such as Time Division Multiplexing PON (TDM-PON)[223], [224], and Wavelength Division Multiplexing PON (WDM-PON)[225], [226], [227].

- Wireless Front End that provides a flexible connection with mobile and fixed end user or device by using Wi-Fi, and LTE [61], [2], [228].

In this chapter, we present an energy-efficient service embedding framework in IoT with cloud and fog processing by using MILP; the framework achieves energy-efficient service embedding in cloud and fog centric IoT for different simultaneous services. We consider the event-driven SOA paradigm in our framework to provide service abstraction of basic services which can be broken down into complex services and exploited by the upper application layer [229], [214].

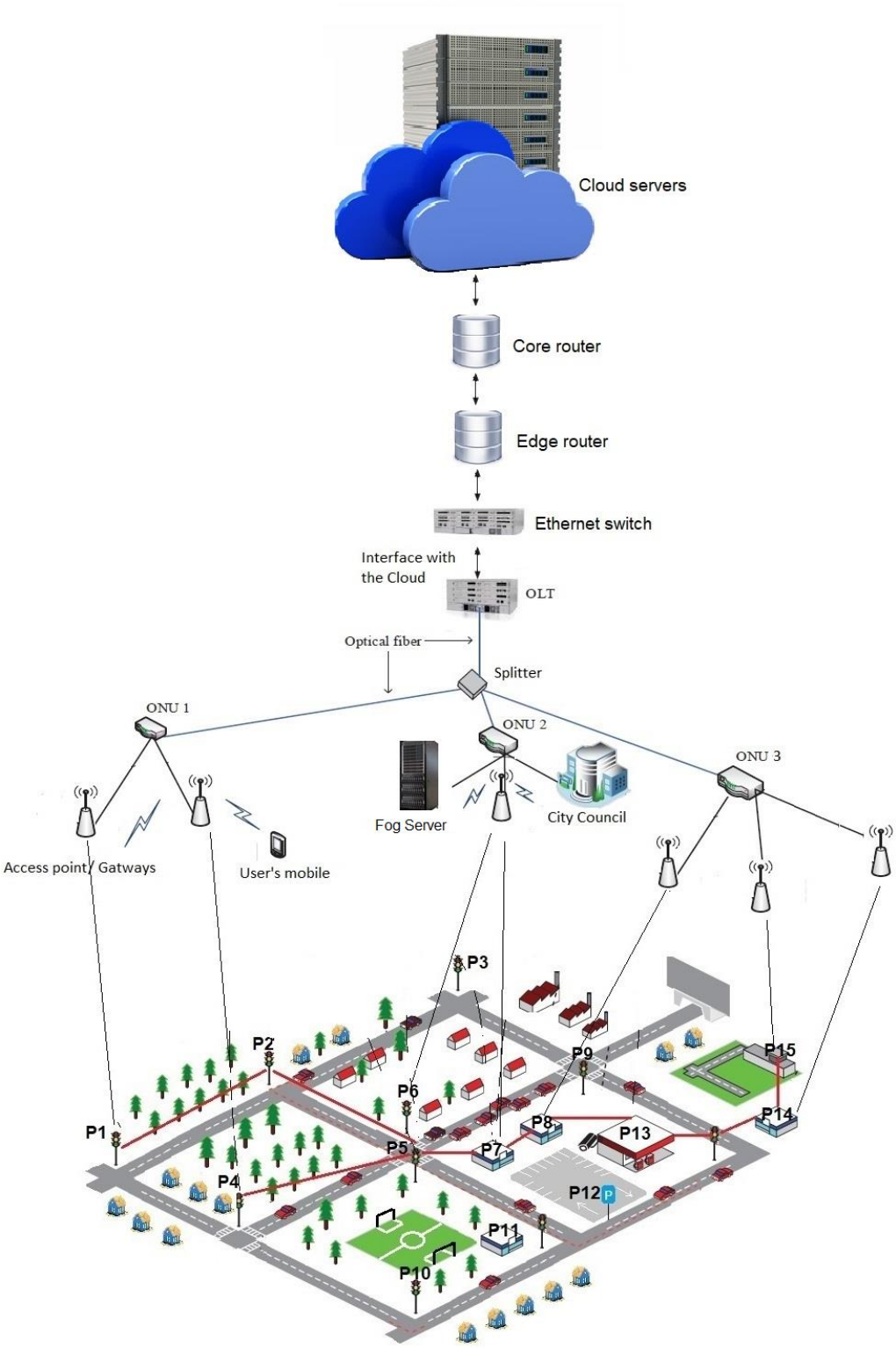


Figure 5-2: Architecture of cloud centric IoT network.

Our framework considers an IoT, fog, and cloud setting in a smart city as shown in Figure 5-2. The IoT layer which consists of IoT nodes is connected to the Wi-Fi network through the Access Points (APs); each AP has been linked with an optical network with corresponding ONU through the optical line. The ONU's provide the

linkage between the IoT network and the fog, and the OLT through the splitter. The OLT has an interface to the cloud through Ethernet switch, edge router, and core router, where the cloud is considered in our architecture as a core node and connected to access networks through a metro network consisting of an Ethernet switch, edge router, and core router. Besides, the service requests are represented in our framework by a set of BPs.

5.3 The framework of energy-efficient service embedding in IoT with cloud and fog processing

We formulate the embedding problem by developing a MILP model with which to select the optimal nodes for processing with the objective function of minimising total power consumption.

5.3.1 Framework Definitions

Before introducing the framework, we define the following sets, parameters, and variables:

Sets

B	Set of business processes (BPs) in the virtual layer
V	Set of virtual nodes in each BP
VN_{ia}	Set of neighbours of each virtual node in each BP ($i \in B, a \in V$)
P	Set of IoT nodes in the physical layer
PN_c	Set of neighbours of IoT nodes ($c \in P$)
F	Set of functions supported by IoT nodes
Z	Set of zones in the IoT physical layer

Parameters

V_{ian}^{FUNC} $V_{ian}^{FUNC} = 1$ If virtual node a in BP i requires the function n, $V_{ian}^{FUNC} = 0$ otherwise

V_{iaz}^{ZONE} $V_{iaz}^{ZONE} = 1$ If virtual node a in BP i requires zone z, $V_{iaz}^{ZONE} = 0$ otherwise

V_{ia}^{MI} Processing requirement of the virtual node a in BP i in MIPS

V_{iab}^{TRFIC} Traffic demand between the virtual node pair (a, b) in $BP\ i$ in kb/s

V_c^{PL} Processing latency demand for processing node (c) in ms

P_{cn}^{FUNC} $P_{cn}^{FUNC} = 1$ If IoT node c can provide the function n, $P_{cn}^{FUNC} = 0$ otherwise

P_{cz}^{ZONE} $P_{cz}^{ZONE} = 1$ If the IoT node c is located in zone z, $P_{cz}^{ZONE} = 0$ otherwise

P_c^{MI} Processing capability of the IoT node c in MIPS

P_{ef}^{DIST} Distance between the neighbouring IoT nodes pair (e, f) in meters

P_c^{IDLECP} Idle MCU power in each node c in mW

P_c^{MAXCP} Maximum processing power consumption of node c in mW

P_c^{IDLETP} Idle network power consumption in each node c in mW

E_{ef}^{PBT} Energy per bit for each IoT link (e, f) in mW/kbps

P_c^{PUE} Power Usage Effectiveness for node c

M Large number ($= 10^8$)

N Maximum number of physical embedding node for each virtual node.

P_c^{CPTY} Node traffic capacity for each IoT node (c) in Mbps

F_{ef}^{TR} Transmit amplifier factor for each link (e, f) in pW/bps/m^2

Variables

I_{iac}^{NE} $I_{iac}^{NE} = 1$ If virtual node a in BP i has been embedded in IoT node c, $I_{iac}^{NE} = 0$ otherwise

I_{iac}^{FP} Utilization of virtual node a in BP i in the embedded in IoT node c.

I_{iacn}^{F} $I_{iacn}^{F} = 1$ If IoT node c has the function n required by virtual node a in BP i, $I_{iacn}^{F} = 0$ otherwise

I_{iacz}^{Z} $I_{iacz}^{Z} = 1$ If IoT node c is located in zone z required by virtual node a in BP i, $I_{iacz}^{Z} = 0$ otherwise

I_{iabcd}^{LE} $I_{iabcd}^{LE} = 1$ If the neighbouring virtual nodes (a, b) in BP i have been embedded in IoT nodes (c, d), $V_{iabcd}^{LE} = 0$ otherwise

R_{cd}^{TRFP} Embedded traffic demand between IoT nodes (c, d) in kbps

R_{cdef}^{ROUTE} Traffic between IoT nodes (c, d) traversing the neighbouring IoT nodes (e, f) in kbps

I^R_{cdef} $I^R_{cdef} = 1$ If the traffic demand between IoT nodes (c, d) traverses neighbouring IoT nodes (e, f), $I^R_{cdef} = 0$ otherwise

R^{TRFL}_{ef} Traffic between neighbouring IoT nodes (e, f) in Mbps

I^{PM}_c $I^{PM}_c = 1$ If the processing module of IoT node c is powered on, $I^{PM}_c = 0$ otherwise

I^{TM}_c $I^{TM}_c = 1$ If the network module of IoT node c is powered on, $I^{TM}_c = 0$ otherwise

P^{PL}_c Processing latency in node c.

TPP Total processing power in the network in mW

TNP Total network power in the network in mW

5.3.2 Framework objective function

This embedding scenario has an objective function which minimises the total power consumption as follows:

Objective: ***minimize*** TNP + TPP (5.1)

where TPP is total processing power for embedding without the processing splitting scenario, and is given by:

$$TPP = \sum_{c \in P} P_c^{PUE} \cdot I_c^{PM} \cdot P_c^{IDLECP} + \sum_{c \in P} \sum_{i \in B} \sum_{a \in V} P_c^{PUE} \cdot I_{iac}^{NE} \cdot P_c^{MAXCP} \cdot \frac{V_{ia}^{MI}}{P_c^{MI}} \quad (5.2)$$

where P_c^{PUE} is power usage effectiveness for each node, $I_c^{PM}$is a binary variable that indicates the active processing module in IoT node, $P_c^{IDLECP}$is the idle processing power parameter of IoT node c in mW, I_{iac}^{NE} is a binary variable which indicates that virtual node a in BP i has been embedded in IoT node c, P_c^{MAXCP} is the parameter of maximum processing power consumption for node c in mW, V_{ia}^{MI} is the parameter of processing requirement of the virtual node a in BP i in mega instruction per second, and P_c^{MI} is the parameter of processing capacity of the node c in mega instruction per second.

The TPP for embedding with the processing splitting scenario is given by:

$$P^{TPP} = \sum_{c \in P} P_c^{PUE} \cdot I_c^{PM} \cdot P_c^{IDLECP} + \sum_{c \in P} \sum_{i \in B} \sum_{a \in V} P_c^{PUE} \cdot I_{iac}^{FP} \cdot P_c^{MAXCP} \quad (5.3)$$

where P_{iac}^{FP} is the fraction variable which indicates that the utilisation of the virtual node a in BP i has been embedded in IoT node c.

The network power consumption is given as:

$$TNP = \sum_{e \in P} P_e^{PUE} \cdot I_e^{TM} \cdot P_e^{IDLETP} + 2 \cdot \sum_{e \in P} \sum_{f \in PN_e} P_e^{PUE} \cdot R_{ef}^{TRFIC} \cdot E_{ef}^{PBT} + \sum_{e \in P} \sum_{f \in PN_e} P_e^{PUE} \cdot R_{ef}^{TRFIC} \cdot (P_{ef}^{DIST})^2 \cdot F_{ef}^{TR} \quad (5.4)$$

where I_e^{TM} is a binary variable that indicates the active network module in IoT node, P_e^{IDLETP} is the idle network power parameter of IoT node e, R_{ef}^{TRFIC} is a variable that describes the traffic between neighbouring IoT nodes (e, f) in Mbps, E_{ef}^{PBT} is the energy per bit for each IoT link (e, f) in mW/Mbps, P_{ef}^{DIST} is the distance between the neighbouring IoT nodes pair (e, f) in metres, and F_{ef}^{TR} is the transmit amplifier factor [64] for each IoT link (e, f) in mW/Mbps.m^2.

5.3.3 Framework Constraints

The framework performs the embedding operation through three parts as follows:

5.3.3.1 Embedding of virtual nodes without processing splitting

$$\sum_{c \in P} I_{iac}^{NE} = 1 \qquad (5.5)$$

$$\forall\, i \in B\,, \qquad \forall\, a \in V$$

$$\sum_{a \in V} I_{iac}^{NE} \leq 1 \qquad (5.6)$$

$$\forall\, i \in B\,, \forall c \in P$$

Constraints (5.5) and (5.6) ensure that each virtual node in a BP is embedded in a single IoT node only and state that each IoT node is not allowed to host more than one virtual node in each BP, where P_{iac}^{NE} is the binary variable which indicates that the virtual node a in BP i has been embedded in IoT node c.

$$\sum_{i \in B} \sum_{a \in V} V_{ia}^{MI} \cdot P_{iac}^{NE} \leq P_c^{MI} \qquad (5.7)$$

$$\forall c \in P$$

Constraint (5.7) represents the processing capacity constraint; it ensures that the embedded processing workloads in the physical node do not exceed the processing capacities.

$$P_{iac}^{NE} \cdot V_{ian}^{FUNC} = V_{iacn}^{FI} \qquad (5.8)$$

$$P_{cn}^{FUNC} >= V_{iacn}^{FI} \quad (5.9)$$

$$\forall i \in B,\ \forall a \in L,\ \forall c \in P,\ \forall n \in F$$

Constraints (5.8) and (5.9) ensure that the required function of each virtual node in BP is provided by its hosting IoT node.

$$P_{iac}^{NE} \cdot V_{iaz}^{ZONE} = V_{iacz}^{ZI} \quad (5.10)$$

$$P_{cz}^{ZONE} \geq V_{iacz}^{ZI} \quad (5.11)$$

$$\forall i \in B, \forall a \in V, \forall c \in P, \forall z \in Z$$

Constraints (5.10) and (5.11) ensure that the required zone of each virtual node in BP is matched by the zone of the hosting IoT node.

5.3.3.2 Embedding of virtual nodes with processing splitting

$$\sum_{c \in P} I_{iac}^{NE} \geq 1 \quad (5.12)$$

$$\forall i \in B, a \in V$$

$$\sum_{c \in P} I_{iac}^{NE} \leq N \quad (5.13)$$

$$\forall i \in B, a \in V$$

Constraint (5.12) ensures that each virtual node in a BP is embedded in at least one or more physical nodes, while constraint (5.13) ensures that each virtual node in a BP is embedded in no more than *N* physical nodes.

$$\sum_{c \in P} P_c^{MI} \cdot I_{iac}^{NE} \geq V_{ia}^{MI} \quad (5.14)$$

$$\forall i \in B, a \in V$$

Constraint (5.14) is a processing constraint; it ensures that the embedded virtual nodes' workloads in the physical node do not exceed the processing capacities.

$$\sum_{c \in P} P_c^{MI} \cdot I_{iac}^{FP} = V_{ia}^{MI} \quad (5.15)$$

$\forall i \in B, a \in V$

Constraint (5.15) creates the fraction of processing of the embedded virtual nodes workloads in the physical node.

$$I_{iac}^{FP} \cdot V_{ia}^{MI} \leq I_{iac}^{NE} \cdot P_c^{MI} \quad (5.16)$$

$\forall i \in B, a \in V, c \in P$

Constraint (5.16) ensures that the embedded virtual nodes' workloads in the physical node do not exceed the processing capacities.

$$\sum_{i \in B} \sum_{a \in V} I_{iac}^{FP} \leq 1 \quad (5.17)$$

$\forall c \in P$

Constraints (5.17) ensures that the processing utilisation of each processing node is not more than 100%.

$$\sum_{a \in V} I_{iac}^{NE} \leq 1 \quad (5.18)$$

$\forall i \in B, \forall c \in P$

Constraint (5.18) states that each IoT node is not allowed to host more than one virtual node in each BP. This is considered as a coexistence constraint that is not used in all scenarios such as controller node virtualisation.

$$\sum_{i \in B} I_{iac}^{NE} \geq I_c^{PM} \quad (5.19)$$

$$\sum_{i \in B} P_{iac}^{NE} \leq I_c^{PM} \cdot M \quad (5.20)$$

$\forall c \in P, a=2$

Constraint (5.19) and (5.20) create a binary variable as a node processing indicator that is equal to 1 when node *c* has embedded the processing of the controller virtual node.

Constraints (5.8 – 5.11) ensure that the required function and zone of each virtual node in BP is matched by the function and zone of the hosting IoT node.

5.3.3.3 Embedding of virtual links

$$I_{iac}^{NE} + I_{ibd}^{NE} = X_{iabcd}^{LE} + 2 \cdot I_{iabcd}^{LE} \quad (5.21)$$

$$\forall\, i \in B\,, \forall a \in V, \forall b \in VN_{ia} : a \neq b, \forall\, c, d \in P: c \neq d$$

Constraint (5.21) generates a binary variable P_{iabcd}^{LE}that indicates each neighbouring virtual nodes pair (a and b) in BP i which is also connected in the embedding IoT nodes (c and d), where X_{iabcd}^{LE} is an auxiliary variable.

$$\sum_{i \in B} \sum_{a \in L} \sum_{b \in LNB_{i\,a}} I_{iabcd}^{LE} \cdot V_{iab}^{TRFIC} = R_{cd}^{TRFP} \quad (5.22)$$

$$\forall\, c, d \in P: c \neq d$$

Constraint (5.22) generates the path's traffic matrix R_{cd}^{TRFP}, where V_{iab}^{TRFIC} is the traffic demand between the virtual node pair (a, b) in $BP\ i$ in Mbps.

$$\sum_{f \in PN_e} R_{cdef}^{ROUTE} - \sum_{f \in PN_e} R_{cdfe}^{ROUTE} \begin{cases} R_{cd}^{TRFP} & if\, e = c \\ -R_{cd}^{TRFP} & if\ e = d \\ 0 & otherwise \end{cases} \quad (5.23)$$

$$\forall\, c, d, e \in P: c \neq d\ and\ e \neq f$$

Constraint (5.23) represents the flow conservation constraint for the traffic flows in the network.

$$\sum_{c \in P} \sum_{d \in P} R_{cdef}^{ROUTE} = R_{ef}^{TRFL} \quad (5.24)$$

$$\forall\, e \in P\,, \forall\, f \in PN_e$$

Constraint (5.24) generates the link's traffic matrix between the neighbouring nodes e and$f$.

$$\sum_{f \in PN_e} R_{ef}^{TRFL} \leq P_e^{CPTY} \quad (5.25)$$

$\forall\, e \in P$

Constraint (5.25) ensures that the total traffic flows of the IoT node e do not exceed node capacity, i.e. 10 Mbps.

$$R_{cdef}^{ROUTE} \geq I_{cdef}^{R} \tag{5.26}$$

$$R_{cdef}^{ROUTE} \leq I_{cdef}^{R} \cdot M \tag{5.27}$$

$\forall\, c, d, e \in P, \forall f \in PN_e : c \neq d\,, e \neq f$

Constraints (5.26) and (5.27) build a path between the embedding nodes c and d through the neighbouring IoT nodes e and$f$, where $I_{cdef}^{R} = 1$ if there is a traffic path between the IoT nodes c and d which passes through the neighbouring IoT nodes e and$f$, where M is a large enough unitless number to ensure that P_{cdef}^{RI} = 1 when $P_{cdef}^{ROUTE}$is greater than zero.

$$\sum_{f \in PN_e} I_{cdef}^{R} \leq 1 \tag{5.28}$$

$\forall c \in P, \forall d \in P, \forall e \in P$

Constraint (5.28) ensures that traffic splitting is prevented for each path between the embedding nodes c and d, such that the maximum number of physical links between neighbouring IoT nodes e and f is one.

$$\sum_{c \in P} \sum_{d \in P} \sum_{f \in PNB_e} I_{cdef}^{R} \geq I_{e}^{TM} \tag{5.29}$$

$$\sum_{c \in P} \sum_{d \in P} \sum_{f \in PNB_e} I_{cdef}^{R} \leq I_{e}^{TM} \cdot M \tag{5.30}$$

$\forall e \in P$

Constraints (5.29) and (5.30) build a network node e if that IoT node is chosen to send/receive traffic at least for one link or more , where M is a large enough unitless number to ensure that I_e^{TM}=1 when $\sum_{c\in P}\ \sum_{d\in P}\sum_{f\in PNB_e}\ I_{cdef}^{R}$ is greater than zero.

The processing latency has been estimated in our framework based on the queuing theory. We considered three levels of delay-sensitive applications, where each level has a specific processing latency constraint.

The model has a latency constraint where processing latency in node c should be less than the processing latency requested, as seen in the following:

$$V_c^{PL}\ >= \frac{1}{(\mu_c^{NODE}-\lambda_c^{NODE})} \tag{5.31}$$

$$\forall c \in P$$

Since we are using linear programming, (5.31) must be converted into a linear format; to facilitate that, we reverse the equation to estimate the mean latency in our framework.

$$P_c^{PL}\ = P_c^{PMI} - \sum_{i\in B}\sum_{a\in V} P_{iac}^{FP}\ \cdot P_c^{PMI} \tag{5.32}$$

$$P_c^{PL}\ \geq I_c^{P}\ /V_c^{PL} \tag{5.33}$$

Constraint (5.32) estimates the processing in node c where P_c^{PL} is a variable which estimates the remaining processing capacity of node c, I_c^P indicates that node c has embedded a virtual processing demand, and V_c^{PL} is a parameter of processing latency demand per mega instruction for node c in ms.

5.4 Results and Evaluations

The framework considers a smart city scenario where the physical layer is composed of 30 IoT nodes which are distributed across a city district of an area spanning 1 km × 1 km. The following considerations are made:

- Nodes with different processing capabilities are distributed in the physical network. The processing capabilities in three levels (IoT, fog, and cloud) in

terms of Mega Instructions per Second (MIPS) are shown in Table 5-1[78]. We considered the same type of server in both the fog and cloud, although a greater number of servers are utilised by the cloud.

Table 5-1: Processing modules specifications and power consumption in active mode

CPU Type	**CPU CLK**	**MIPS**	**Max Power**	**PUE**	**Location**
IoT-with low processing capacity	48 MHz	100	20 mW	1	IoT
IoT-with higher processing capacity	400 MHz	856	110 mW	1	IoT
Fog	1.86 GHz	7500	40 W	1.3	Fog
Cloud	1.86 GHz	7500	40 W	1.4	Cloud

- The power consumption of the wireless network elements considered is composed of the power due to the idle mode, electronics, and transmitter power amplifier. In the optical network the corresponding energy per bit as shown in Table 5-2.

Table 5-2: Network modules specifications and power consumption in active mode [230]

Network elements	Energy per bit	Capacity	PUE	Location
Ethernet SW, Edge and core router	28 nJ/b	40 Gbps	1.5	Core network
Optical Line Terminal	12 nJ/b	40 Gbps	1.5	Optical backbone
Optical Network Unit	5 nJ/b	1 Gbps	1.5	Optical backbone
Shared Wi-Fi Access Point	100 nJ/b	100 Mbps	1.5	Wireless front end

- There is a set of seven different functions: three sensing functions (camera, climate sensor, and motion sensors), one control function and three actuation functions (alarms, display screens, and traffic lights). Each IoT node can provide four functions only from this set while the virtual node requests are for one function only.
- The processing demand of the controller virtual node is 1000 MIPS.
- The traffic demand of virtual links is 1 Mbps.

- The IoT nodes are connected to the AP via 10 Mbps transceiver modules with the IEEE 802.11 stack [231].
- There is a set of five geographical zones that represent the sub-districts of the smart city. The IoT nodes are distributed randomly and uniformly over these zones and each virtual node requests a location in one of these five zones.
- We have adopted a PUE of 1.3, 1.4 and 1.5 for the fog, cloud and network equipment respectively while considering the PUE of the IoT to be equal to 1.

In this work, we considered the processing latency only as the traffic and propagation latency are very small compared to the processing latency.

The processing latency on the physical node *c* is given by:

$$Processing_{c}^{Latency} = \frac{1}{(\mu_c^{NODE} - \lambda_c^{NODE})} \quad (5.34)$$

$$\forall c \in P$$

where μ_c^{NODE} is the processing capacity of node *c* and λ_c^{NODE} is the processing demand that embedded on node *c* with given processing capacity for the cloud and IoT and the processing demands of 1000 MIPS for each BP:

The average delay per MIPS in the cloud:

$$Processing_{Cloud}^{Latency\ of\ 1000\ MIPS} = \frac{1}{(22500-1000)} = 46\ \mu \text{ second /MIPS}$$

Hence the processing latency for 1000MIPS of one BP demands in the cloud server is **46** milli seconds.

On the other hand, the processing latency for the same processing demand on the IoT after splitting of the processing between two IoT nodes:

$$Processing_{IoT}^{Latency\ of\ 1000\ MIPS} = \frac{1}{(856-500)} = 2.8 \text{ millisecond /MIPS}$$

Hence the processing latency for 1000 MIPS of one BP demand on the IoT nodes is **1.4** seconds.

The cloud and fog are higher than the IoT in processing capacity and power consumption due to the processing architecture.

Distinctly the best placement for the processing demand from the point of view of power consumption is in the IoT node as thus reduce the network power consumption. The placement of processing in the IoT nodes is only possible if two conditions are met:

- The processing latency requirement is met
- The number of IoT nodes are enough meet the split processing demands for the given BPs.

We studied the embedding of BPs with the objective of minimising power consumption and evaluated the impact of processing splitting and the coexistence constraint for the following three scenarios:

- Scenario A, where the model ensures that the processing in any node happens with an average processing latency constraint of less than 1 second i.e. the security application that uses camera for face or plate number recognition [232].
- In scenario B, the model ensures that the processing happens in any node with a constraint on average processing latency less than 2 seconds i.e. traffic light control and pedestrian traffic monitoring services [233].
- In scenario C, the model ensures that the processing happens in any node with a constraint on average processing latency less than 5 seconds i.e. car parking services [234].

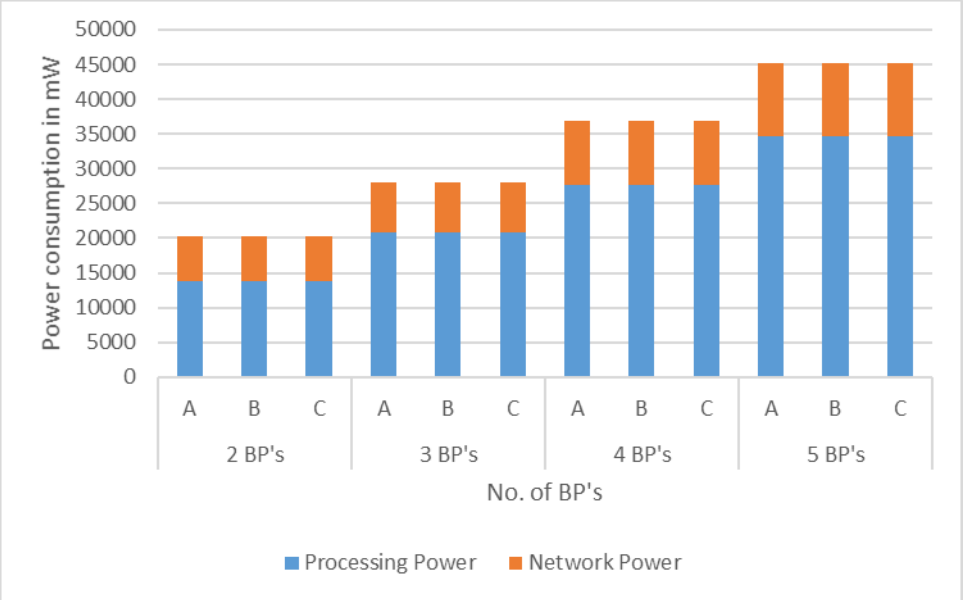


Figure 5-3: Power consumption of service embedding without processing splitting and with coexistence constraint.

Figure 5-3 shows the processing and network power consumption of three scenarios of service embedding without processing splitting capability and without coexistence constraint. The results show that the power consumed by the processing modules contributes approximately 73% (on average) of the total power consumption of the network for all scenarios.

The results illustrate that the highest power consumption in all scenarios, which show that the whole processing has been placed in the fog and cloud servers to satisfy the processing latency constraint; consequently, this consumes higher processing and traffic power.

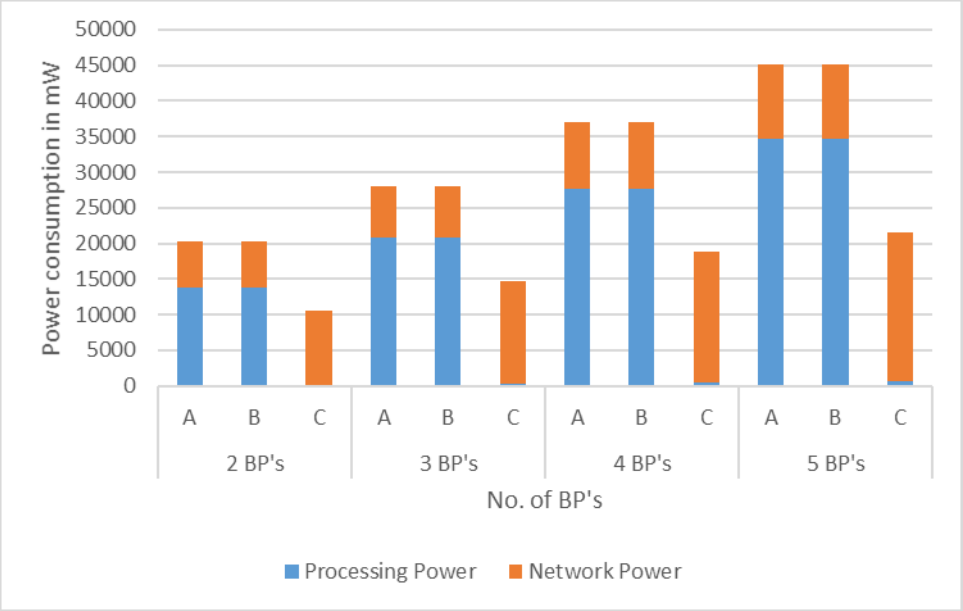


Figure 5-4: Power consumption of service embedding with processing splitting and with coexistence constraint.

The results in Figure 5-4 illustrate the power consumption of service embedding scenarios with the ability of the processing splitting and coexistence constraint. The results also show that scenarios A and B have the same power consumption level compared with Fig. 5.3.

Scenario C has an average power saving of 48% compared with scenarios A and B. The power saving results due to the ability to embed a part of processing demands in the IoT nodes, which means low processing and traffic power consumption with lower PUE values.

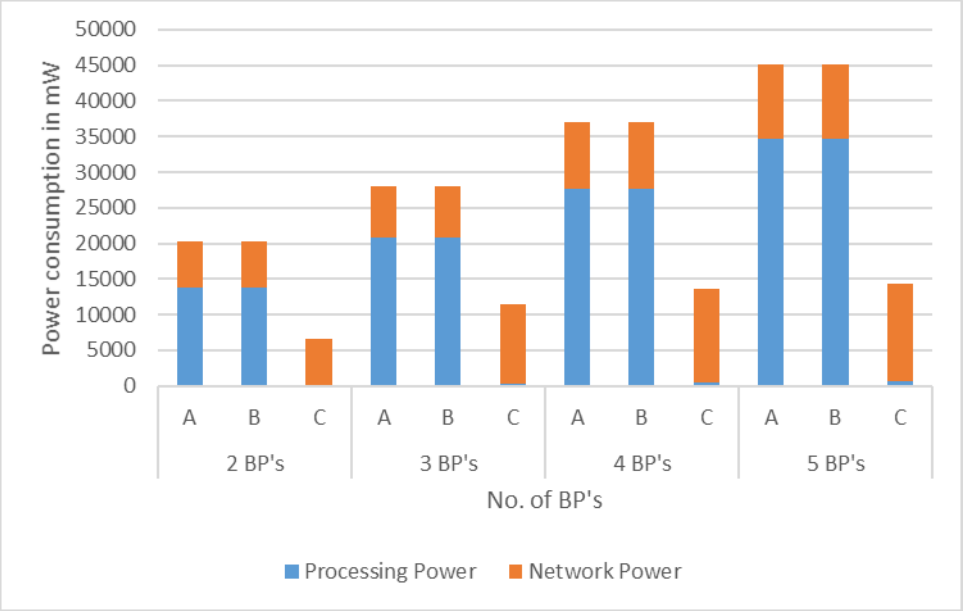


Figure 5-5: Power consumption of service embedding with processing splitting and without coexistence constraint.

The results in Figure 5-5 display the power consumption of the service embedding with processing splitting without coexistence constraint. The results also show that scenarios A and B have the same power consumption level, while scenario C has an average power saving of 64% compared with scenarios A and B. The power saving here is due to the ability to embed the processing node in the same physical node of the sensor or actuator, and consequently there is no traffic between these nodes due to the embedding in the same node.

The power saving is proportional to the processing latency constraint. The higher processing latency constraint has the impact of higher power saving as it grants more IoT nodes to embed the processing. As such, it is considered an energy-efficient processor but has limited processing capacity compared with the fog and cloud.

The processing in IoT nodes consumes a small amount of power due to the energy-efficient architectural processing of the IoT nodes.

The PUE values related to the data centre's equipment for the cloud and fog are considered as an important factor that consumes a greater amount of power than the IoT nodes. Another factor that effects the power consumption is the traffic modules

and equipment. The traffic from the IoT nodes and the cloud and fog needs to be routed through the wireless and PON and core network equipment.

The coexistence constraint increases the power consumption. Under coexistence constraints, the virtual nodes of the BP cannot coexist in the same IoT node to improve the resilience of the BPs under single node failure of the framework. Indeed, without coexistence constraint, the model embeds the controller node either in the same node of the traffic generator (i.e. the sensor node) or the actuator node to reduce the traffic overhead and embeds the highest number of virtual nodes in the minimal number of IoT nodes.

5.5 Summary

This chapter has investigated the energy efficiency of service embedding in a smart city setting. We extended our IoT network architecture by using cloud and fog processing. We developed a MILP framework to optimise the allocation of the processing demands, which minimises the processing and network power consumption.

We evaluated the total power consumption for three levels of processing latency constraints (i.e. different scenarios). The results show that the minimum power consumption is seen in the service embedding scenario with the highest processing latency constraint (i.e. in 5 seconds) compared with the service embedding scenario with the lowest processing latency constraint (1 second).

We also investigated the impact of splitting the processing, on the power consumption. The results show that it is possible to save energy by processing splitting through optimising the traffic distribution in the IoT network while satisfying the service requirements. Finally, we investigated the impact of the coexistence constraint on reducing the power saving in service embedding in the processing splitting scenario.

Chapter 6
Resilient Energy-Efficient Service Embedding in Smart Buildings with Latency Minimisation

6.1 Introduction

The IoT concept promises a countless number of services ranging from those in smart homes to the automation of industries and public utilities. However, the growth of these deployments has posed a significant challenge: how can we build such deployments in a highly resilient manner? The IoT nodes are possibly prone to unexpected failures and malicious attacks, i.e. various types of damage, unreliable wireless connections, limited transmission power, computing ability, and storage space. The IoT paradigm consists of a heterogeneous combination of Internet-connected devices. Further, traffic routing in IoT networks mainly relies on Routing Protocol for Low-Power and Lossy Networks (RPL). The RPL has been designed to find a single route between the source and the destination nodes [235]. The RPL is considered an energy-efficient routing protocol but has an impact on the services delivered by the networks, such as intermittent node's faults or dropped radio links due to energy saving or a change in the network connectivity in addition to the vulnerability of attacks[236].

This chapter introduces a generic MILP model that has been developed to minimize the power consumption due to both processing and the traffic flow through the network to minimize the end to end data delivery time with resilient embedding. We investigate various resilience schemes for IoT nodes and traffic and evaluate the performance and the implications of these schemes in smart building settings, such as the data delivery time and energy consumption. We formulate the problem of finding the optimal set of IoT nodes and links to embed BPs into the IoT layer as an optimisation problem, with an objective function that aims to minimise both the total power consumption and the traffic latency.

6.2 Proposed Architecture

We developed a framework that enhances the resilience of a service embedding IoT network in the smart building setting proposed in Chapter 4. This framework aims to structure a network with an acceptable level of fault tolerance and implies an ability to restore from a node or link failure.

Our framework is based on a probabilistic approach that assumes that the idiom of k-connected networks denotes that the network is preserving its node connectivity after the removal of no more than one node or link from the network. The framework proposes multilevel resilience schemes, where each probable type of failure (i.e. sensor, controller, or link failure) requires an appropriate level of failure recovery. We evaluated the proposed resilience levels by considering their impact on the end-to-end service delay and energy consumption. The proposed resilience levels are as follows:

6.2.1 Resilient service embedding with node coexistence constraint.

We considered the service embedding with a coexistence constraint as the basic level of resilience. This scheme is considered to be the basic solution for a network with a probable temporary failure, i.e. data collision or packet drop.

This resilience scheme is based on a single path between the source and the destination nodes, where the source node is insured in terms of recovering the lost packets by retransmitting them till an acknowledgement is received from the destination node. This scheme has the disadvantages of additional transmission overhead, high network congestion, and poor effects on unreliable data transmission.

6.2.2 Resilient service embedding with sensor–actuator node redundancy.

To enhance the resilience of IoT networks, another solution has been proposed. It introduces redundant nodes and links for the sensor and actuator nodes. The redundancy scheme enhances the infrastructure's resilience against service failure or disruptive attacks. Network architectures with partial redundant components were used to investigate the addition of some nodes and links as redundant ones to

enhance the performance of the provisioned services. Consequently, we considered the redundant sensing and actuating nodes for accuracy and data fidelity in addition to the resilience concern.

6.2.3 Resilient service embedding with all-node redundancy.

In many services, resilience has significant importance, such as fire protection and security services in public buildings. As the cost of the service components (e.g. nodes and energy consumption) is a non-substantial concern, a new feasible scheme based on the allocation of redundant components for all the nodes enables end-to-end traffic routing with multiple paths capability.

6.2.4 Resilient service embedding with traffic redundancy.

This scheme is related to traffic resilience and is based on setting up multiple paths between the source and the destination nodes. One of these paths is considered the main or primary path to route the traffic between the nodes, while one or more other paths are considered the alternative or backup paths. These paths are used to recover from a traffic failure of the primary path and are sustained by sending a ‘Keep-alive’ signal continuously over them. When a primary path has a failure, the intermediate node will send back the data packet to the source node and send a failure report to the destination node. As a result, the source and the destination nodes will remove the failed path information from the routing table and switch the traffic to an alternative path.

6.2.5 Resilient service embedding with traffic replication.

This scheme fulfils the requirement of resilient traffic by sending multiple replicas of the data over selected multiple paths from the source node to the destination node. This technique has the advantages of a high packet delivery ratio with a low data delivery time, and there is no need for signalling the state maintenance between the source node and the destination node, because even in the case of a partial data packet loss, the destination node can recover the packet from the other copies of the packet. Replication has achieved high resilience but at the cost of high energy consumption that arises because of the traffic overhead at each node along with the network.

6.2.6 Resilient service embedding with traffic splitting.

Here, we propose our novel technique by traffic splitting from the source node to the destination node in two paths, where each path will route 50% of the data traffic, and the 'Keep-alive' signal will be redirected on the same path; when a failure is encountered on one path, the source will resend the undelivered data, which will not exceed 50% of the original data, of the failed path on the second path; consequently, we will save both energy and delivery time. We proposed an equal rate of splitting (i.e. 50% and 50%) due to the consideration that all the network's links have the same level of reliability and availability, such that, all the links have the same level of the priority to route the traffic of data. We propose the use of a braided multipath technique in our framework; in this technique, the alternative nodes partially overlay the nodes of the primary path to avoid service blockage.

6.3 Framework of Resilient Energy-Efficient Service Embedding in IoT Networks

In this section, we introduce the framework developed to embed services in IoT networks. This framework is based on a MILP optimisation model with the objective of minimising the total energy consumption and the traffic mean latency of the service embedding in IoT networks and enhancing the node/traffic resilience level.

6.3.1 Framework definitions

Before introducing the framework, we define the following sets, parameters, and variables:

Sets

B	Set of business processes (BPs) in the virtual layer
V	Set of virtual nodes in each BP
VN_{ia}	Set of neighbours of each virtual node in each BP $(i \in B, a \in V)$
P	Set of IoT nodes in the physical layer
PN_c	Set of neighbours of IoT nodes $(c \in P)$

F	Set of functions supported by IoT nodes
Z	Set of zones in the IoT physical layer
λ	Set of arrival rates
W_j	Set of mean latency per arrival rate ($j \in \lambda$) in ms per packet

Parameters

V_{ian}^{FUNC}	$V_{ian}^{FUNC} = 1$ If virtual node a in BP i requires the function n, $V_{ian}^{FUNC} = 0$ otherwise
V_{iaz}^{ZONE}	$V_{iaz}^{ZONE} = 1$ If virtual node a in BP i requires zone z, $V_{iaz}^{ZONE} =$ 0 otherwise
V_{ia}^{MCU}	Processing requirement of the virtual node a in BP i in MHz
V_{ia}^{RAM}	Memory requirement of the virtual node a in BP i in kB
V_{iab}^{TRFIC}	Traffic demand between the virtual node pair (a, b) in $BP\ i$ in kb/s
P_{cn}^{FUNC}	$P_{cn}^{FUNC} = 1$ If IoT node c can provide the function n, $P_{cn}^{FUNC} = 0$ otherwise.
P_{cz}^{ZONE}	$P_{cz}^{ZONE} = 1$ If the IoT node c is located in zone z, $P_{cz}^{ZONE} =$ 0 otherwise.
P_c^{MCU}	Processing capability of the IoT node c in MHz.
P_c^{RAM}	Memory capability of the IoT node c in kB.
P_{ef}^{DIST}	Distance between the neighbouring IoT nodes pair (e, f) in meters.
P_c^{IDLECP}	Idle processor power in each IoT node c in mW.
P_c^{MAXCP}	Maximum processor power consumption in each IoT node c in mW.
P_c^{IDLETP}	Idle network power consumption in each IoT node c in mW.
E_{ef}^{PBT}	Energy per bit for each IoT link (e, f) in mW/kbps.
M	Large number ($= 10^8$).
P_e^{CAPT}	Link capacity for each IoT node (e) in kbps.
F_{ef}^{TR}	Transmit amplifier factor for each IoT link (e, f) in mW/kbps/m^2.

Variables

I_{iac}^{NE} I_{iac}^{NE} is node embedding indicator, $I_{iac}^{NE} = 1$ If virtual node a in BP i has been embedded in IoT node c, $I_{iac}^{NE} = 0$ otherwise.

I_{iacn}^{F} I_{iacn}^{F} is function embedding indicator, $I_{iacn}^{F} = 1$ If IoT node c has the function n required by virtual node a in BP i, $I_{iacn}^{F} = 0$ otherwise.

I_{iacz}^{Z} I_{iacz}^{Z} is zone embedding indicator, If IoT node c is located in zone z required by virtual node a in BP i,$I_{iacz}^{Z} = 0$ otherwise.

I_{iabcd}^{LE} I_{iabcd}^{LE} is link embedding indicator, $I_{iabcd}^{LE} = 1$If the neighbouring virtual nodes (a, b) in BP i have been embedded in IoT nodes (c, d), $I_{iabcd}^{LE} = 0$ otherwise.

X_{iabcd}^{XOR} Dummy binary variable

R_{cd}^{TRFP} Embedded traffic demand between IoT nodes (c, d) in kbps.

$R1_{cdef}^{TR}$ Primary path between IoT nodes (c, d) traversing the neighbouring IoT nodes (e, f) in kbps.

$R2_{cdef}^{TR}$ Secondary path between IoT nodes (c, d) traversing the neighbouring IoT nodes (e, f) in kbps.

I_{cdef}^{R1} Primary path indicator, $I_{cdef}^{R1} = 1$ If the traffic demand between IoT nodes (c, d) traverses neighbouring IoT nodes(e, f), $I_{cdef}^{R1} = 0$ otherwise.

I_{cdef}^{R2} Secondary path indicator, $I_{cdef}^{R2} = 1$ If the traffic demand between IoT nodes (c, d) traverses neighbouring IoT nodes(e, f), $I_{cdef}^{R2} = 0$ otherwise.

R_{ef}^{TRFL1} Traffic between neighbouring IoT nodes (e, f) in kbps.

R_{ef}^{TRFL2} Traffic between neighbouring IoT nodes (e, f) in kbps.

R_{f}^{TRFN} Arrival rate of IoT nodes (f) in kbps.

$LI_{f\ j}^{\mathrm{Lmbda}}$ Lambda indicator for each IoT node (f) with correspondent arrival rate (j) then $LI_{f\ j}^{\mathrm{Lmbda}} = 1$, otherwise 0.

W_f^{NODE} Traffic mean latency for each node (f)in ms.

I_c^{PM} $I_c^{PM} = 1$ If the processing module indicator of IoT node c is powered on, $P_c^{PM} = 0$ otherwise.

I_c^{TM} $I_c^{TM} = 1$ If the network module indicator of IoT node c is powered on, $I_c^{TM} = 0$ otherwise.

TPP Total processing power in the IoT network in mW.

TNP Total network consumption in the IoT network in mW.

TL Total traffic mean latency in traffic the primary path in ms.

6.3.2 Framework objective function

The proposed framework minimises the power consumption and the queuing latency in an IoT network by using the following objective function:

Objective: ***minimise*** $\alpha.TL + \beta.TPP + \gamma.TNP$ (6.1)

where α, β, and γ are the weight values thus used for magnitude and units. The framework selects the traffic value for each link in the network that preserves the low power consumption and the mean traffic latency at feasible values of the arrival rate. To enhance optimality for the power saving and latency minimisation, we used the weights values given in Chapter 4 ($\alpha = 30/\text{ms}$, $\beta = 1/\text{mW}$, and $\gamma = 1/\text{mW}$).

Here, the total traffic latency for the IoT nodes can be calculated as follows:

$$TL = \sum_{f \in P} W_f^{NODE} \quad (6.2)$$

where W_f^{NODE} represents the average waiting time of the packets waiting to be processed for each IoT node in milliseconds according to the queuing theory.

TPP is the total processing power and can be calculated as follows:

$$TPP = \sum_{c \in P} I_c^{PM} \cdot P_c^{IDLECP} + \sum_{c \in P} \sum_{i \in B} \sum_{a \in V} I_{iac}^{NE} \cdot P_c^{MAXCP} \cdot \frac{V_{ia}^{MCU}}{P_c^{MCU}} \quad (6.3)$$

where I_c^{PM} is the binary variable that indicates an active processing module in IoT node *c*, P_c^{IDLECP} is the idle processing power parameter of IoT node *c* in milli watts, I_{iac}^{NE} is a binary variable that indicates that virtual node a in BP *i* has been embedded in IoT node *c*, P_c^{MAXCP} is the parameter of maximum CPU power consumption in each IoT node *c* in milli watts, V_{ia}^{MCU} is the parameter of the processing requirement of virtual node *a* in BP *i* in megahertz, and P_c^{MCU} is the parameter of the processing capability of the IoT node *c* in megahertz. The processing power consumption is considered to follow a linear profile versus the load with idle power consumption.

Here, the network power consumption in the IoT network can be expressed as follows:

$$TNP = \sum_{e \in P} I_e^{TM} \cdot P_e^{IDLETP} + 2 \cdot \sum_{e \in PN} \sum_{f \in PB_e} R_{ef}^{TRFL1} \cdot E_{ef}^{PBT} + 2 \cdot \sum_{e \in PN} \sum_{f \in PB_e} R_{ef}^{TRFL2} \cdot E_{ef}^{PBT} + \sum_{e \in PN} \sum_{f \in PB_e} R_{ef}^{TRFL1} \cdot (P_{ef}^{DIST})^2 \cdot F_{ef}^{TR} + \sum_{e \in PN} \sum_{f \in PB_e} R_{ef}^{TRFL2} \cdot (P_{ef}^{DIST})^2 \cdot F_{ef}^{TR} \quad (6.4)$$

where I_e^{TM} is the binary variable that indicates an active network module in IoT node e, P_e^{IDLETP} is the idle network power parameter of IoT node e, R_{ef}^{TRFL1} and R_{ef}^{TRFL2} indicate the primary and alternative paths' traffic between neighbouring IoT nodes (e, f) in kb/s, E_{ef}^{PBT} represents the energy per bit for each IoT link (e, f) in milliwatts per kilobit per second, P_{ef}^{DIST} denotes the distance between the

neighbouring IoT nodes pair (e, f) in meters, and F_{ef}^{TR} represents the transmit amplifier factor [64] for each IoT link (e, f) in milliwatts per kilobit per second per metre square.

6.3.3 Framework constraints

The proposed framework performs the embedding operation in two parts as follows:

6.3.3.1 Embedding of virtual nodes

$$\sum_{c \in P} I_{iac}^{NE} = 1 \tag{6.5}$$

$$\forall\, i \in B\,, \qquad \forall\, a \in V$$

$$\sum_{a \in V} I_{iac}^{NE} \leq 1 \tag{6.6}$$

$$\forall\, i \in B\,, \forall c \in P$$

Constraint (6.5) ensures that each virtual node in a BP is embedded in a single IoT node only. Constraint (6.6) states that each IoT node is not allowed to host more than one virtual node in each BP. This is considered the coexistence constraint and is not used in all the scenarios, such as controller node virtualisation.

$$\sum_{i \in B} \sum_{a \in V} I_{iac}^{NE} \geq I_c^{PM} \tag{6.7}$$

$$\forall\, c \in P$$

$$\sum_{i \in B} \sum_{a \in V} I_{iac}^{NE} \leq I_c^{PM} \cdot M \tag{6.8}$$

$$\forall\, c \in P$$

Constraints (6.7) and (6.8) build the processing module of IoT node c if this node is chosen for embedding at least one virtual node a in BP i or more, where M is a sufficiently large unitless number to ensure that $P_c^{PMI} = 1$ when $\sum_{i \in B} \sum_{a \in V} P_{iac}^{NE}$ is greater than zero.

$$\sum_{i\in B}\sum_{a\in V} V_{ia}^{MCU} \cdot I_{iac}^{NE} \leq P_c^{MCU} \quad (6.9)$$

$$\forall c \in P$$

$$\sum_{i\in B}\sum_{a\in L} V_{ia}^{RAM} \cdot I_{iac}^{NE} \leq P_c^{RAM} \quad (6.10)$$

$$\forall c \in P$$

Constraints (6.9) and (6.10) represent the MCU and the memory capacity constraints, respectively. They ensure that the embedded MCU and memory workloads in an IoT node do not exceed the processor and memory capacities, respectively.

$$I_{iac}^{NE} \cdot V_{ian}^{FUNC} = I_{iacn}^{F} \quad (6.11)$$

$$P_{cn}^{FUNC} >= I_{iacn}^{F} \quad (6.12)$$

$$\forall i \in B,\ \forall a \in L,\ \forall c \in P,\ \forall n \in F$$

Constraints (6.11) and (6.12) ensure that the required function of each virtual node in BP is provided by its hosting IoT node.

$$I_{iac}^{NE} \cdot V_{iaz}^{ZONE} = I_{iacz}^{Z} \quad (6.13)$$

$$P_{cz}^{ZONE} \geq I_{iacz}^{Z} \quad (6.14)$$

$$\forall i \in B, \forall a \in V, \forall c \in P, \forall z \in Z$$

Constraints (6.13) and (6.14) ensure that the required zone of each virtual node in BP is matched by the zone of the hosting IoT node.

6.3.3.2 Embedding of virtual links

$$I_{iac}^{NE} + I_{ibd}^{NE} = X_{iabcd}^{LE} + 2 \cdot I_{iabcd}^{LE} \quad (6.15)$$

$$\forall i \in B, \forall a \in V, \forall b \in VN_{ia} : a \neq b, \forall c, d \in P : c \neq d$$

Constraint (6.15) ensures that neighbouring virtual nodes a and b of i in B are also connected in embedding IoT nodes c and d. We achieved this by introducing a

binary variable I_{iabcd}^{LE}, which is only equal to 1 if I_{iac}^{NE} and I_{ibd}^{NE} are exclusively equal to 1; otherwise, it is zero, when X_{iabcd}^{LE} is a neglected variable.

$$\sum_{i \in B} \sum_{a \in L} \sum_{b \in LNB_{i\,a}} I_{iabcd}^{LE} \cdot V_{iab}^{TRFIC} = R_{cd}^{TRFP} \tag{6.16}$$

$$c, d \in P: c \neq d$$

Constraint (6.16) generates the path's traffic matrix resulting from embedding virtual nodes a and b into IoT nodes c and d.

6.3.3.3 Retransmission- and replication-based schemes

In this scheme, the proposed framework finds two energy-efficient routes for the traffic between the embedded nodes, namely the primary and alternative routes.

$$\sum_{f \in PN_e} R_{cdef}^{TR1} - \sum_{f \in PN_e} R_{cdfe}^{TR1} \begin{cases} R_{cd}^{TRFP} & if\, e = c \\ -R_{cd}^{TRFP} & if\ e = d \\ 0 & otherwise \end{cases} \tag{6.17}$$

$$\forall\, c, d, e \in P: c \neq d \; and \; e \neq f$$

Constraint (6.17) represents the flow conservation constraint for the traffic flows in the IoT network.

$$\sum_{c \in P} \sum_{d \in P} R_{cdef}^{TR1} = R_{ef}^{TRFL1} \tag{6.18}$$

$$\forall\, e \in P\,, \forall\, f \in PN_e$$

Constraint (6.18) generates a link's traffic matrix between the neighbouring IoT nodes e and f.

$$R_{cdef}^{TR1} \geq I_{cdef}^{R1} \tag{6.19}$$

$$R_{cdef}^{TR1} \leq I_{cdef}^{R1} \cdot M \tag{6.20}$$

$$\forall\, c, d, e \in PN, \forall f \in PB_e : c \neq d\,, e \neq f$$

Constraints (6.19) and (6.20) build the primary path indicator between embedding IoT nodes c and d through neighbouring IoT nodes e and f, where $I_{cdef}^{R1} = 1$ if there is a traffic path between IoT nodes c and d that passes through neighbouring IoT nodes e and f, where *M* is a sufficiently large unitless number to ensure that R_{cdef}^{R1} = 1 when R_{cdef}^{ROUTE1} is greater than zero.

$$\sum_{f \in PB_e} I_{cdef}^{R1} \leq 1 \tag{6.21}$$

$$\forall c, d, e \in PN : c \neq d \; and \; e \neq f$$

Constraint (6.21) ensures that traffic splitting is prevented for each path between embedding IoT nodes c and d, such that the maximum number of physical links between neighbouring IoT nodes *e* and *f* is one.

$$\sum_{f \in PN_e} R_{cdef}^{TR2} - \sum_{f \in PN_e} R_{cdfe}^{TR2} \begin{cases} R_{cd}^{TRFP} & if e = c \\ -R_{cd}^{TRFP} & if \; e = d \\ 0 & otherwise \end{cases} \tag{6.22}$$

$$\forall \; c, d, e \; \in \; PN: c \neq d \; and \; e \neq f$$

Constraint (6.22) represents the flow conservation constraint for the alternative path's traffic flows in the IoT network.

$$\sum_{c \in P} \sum_{d \in P} R_{cdef}^{TR2} = R_{ef}^{TRFL2} \tag{6.23}$$

$$\forall \; e \in PN \; , \forall \; f \in PB_e$$

Constraint (6.23) generates the alternative link's traffic matrix between neighbouring IoT nodes e and f.

$$R_{cdef}^{TR2} \geq I_{cdef}^{R2} \tag{6.24}$$

$$R_{cdef}^{TR2} \leq I_{cdef}^{R2} \cdot M \tag{6.25}$$

$$\forall\, c,d,e \in PN, \forall f \in PB_e : c \neq d\ , e \neq f$$

Constraints (6.24) and (6.25) build the alternative path between embedding IoT nodes c and d through neighbouring IoT nodes e and f, where $R_{cdef}^{R2} = 1$ if there is a traffic path between IoT nodes c and d that passes through neighbouring IoT nodes e and f, where M is a sufficiently large unitless number to ensure that I_{cdef}^{R2} = 1 when R_{cdef}^{TR2} is greater than zero.

$$\sum_{f \in PB_e} R_{cdef}^{R2} \leq 1 \tag{6.26}$$

$$\forall c,d,e \in\ PN\ : c \neq d\ and\ e \neq f$$

Constraint (6.26) ensures that traffic splitting is prevented for each path between embedding IoT nodes c and d, such that the maximum number of physical links between neighboring IoT nodes e and f is one.

$$I_{cdef}^{R1} + I_{cdef}^{R2} \leq 1 \tag{6.27}$$

$$\forall\, c,d,e \in PN, \forall f \in PB_e : c \neq d\ , e \neq f$$

Constraint (6.27) ensures the traffic creation of two distinct paths between embedding IoT nodes c and d such that each path uses different physical links between neighbouring IoT nodes e and f.

$$\sum_{c \in PN} \sum_{d \in PN} \sum_{f \in PB_e} I_{cdef}^{R1} + I_{cdef}^{R2} \geq I_e^{TM} \tag{6.28}$$

$$\sum_{c \in PN} \sum_{d \in PN} \sum_{f \in PB_e} R_{cdef}^{R1} + R_{cdef}^{R2} \leq I_e^{TM} \cdot M \tag{6.29}$$

$$e \in\ PN : c \neq d\ and\ e \neq f$$

Constraints (6.28) and (6.29) build a network module indicator of IoT node e if this IoT node is chosen for send/receive traffic for at least one link or more, where

M is a sufficiently large unitless number to ensure that I_e^{TM} = 1 when $\sum_{c\in PN}\ \sum_{d\in PN}\sum_{f\in PB_e}\ I_{cdef}^{R1} + I_{cdef}^{R2}$ is greater than zero.

$$\sum_{e\in PN_f} R_{ef}^{TRFL1} + R_{ef}^{TRFL2} = R_f^{TRFN} \tag{6.30}$$

$$\forall\, f \in P : e \neq f$$

Constraint (6.30) estimates the arrival traffic for each IoT node.

$$\sum_{f\in P} R_f^{TRFN} \leq CAPACITY \tag{6.31}$$

Constraint (6.31) states that the total traffic flow of the IoT node f should not exceed the node capacity.

$$\sum_{j\in J} LI_{f\ j}^{LMBDA} \cdot j = R_f^{TRFN} \tag{6.32}$$

$$\forall\, f \in P$$

Constraint (6.32) determines the arrival rate for each IoT node.

$$\sum_{j\in J} LI_{f\ j}^{LMBDA} \leq 1 \tag{6.33}$$

$$\forall\, f \in P$$

Constraint (6.33) ensures that each IoT node has no more than one arrival rate indicator.

$$\sum_{j\in J} W_j^{LMBDA} \cdot LI_{f\ j}^{LMBDA} = W_f^{NODE} \tag{6.34}$$

$\forall f \in P$

Constraint (6.34) estimates the traffic delay for each IoT node f based on the product of the lambda indicator and the correspondent latency for this lambda j.

6.3.3.4 Splitting-based schemes

In this section, we propose a traffic splitting-based resilience scheme through the multiple paths concept to reduce the arrival rates through the intermediate nodes; doing so will consequently minimise the delivery time, in addition to enhancing the resilience of the IoT network.

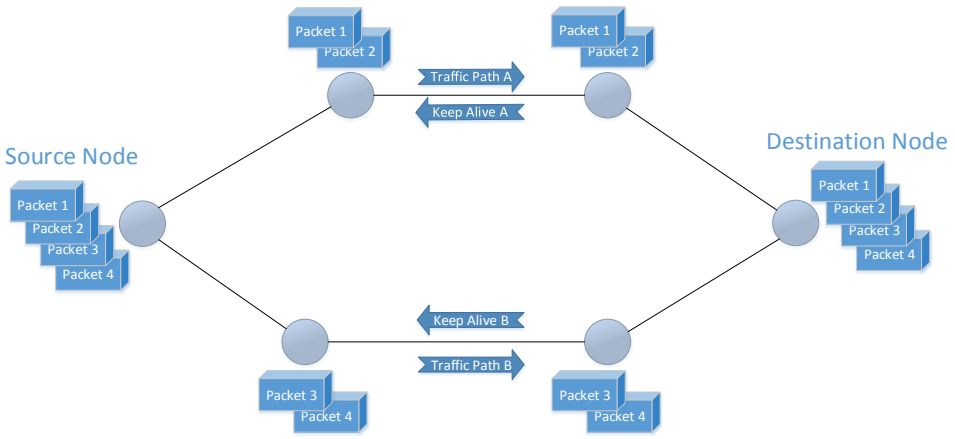


Figure 6-1: Traffic Splitting Scheme

The proposed framework splits the traffic between the source node splits and routes it into two paths (A and B), as shown in Figure 6-1. The source node sends one half of the traffic through path A and the other half through path B to the destination node, and the source node receives a 'Keep-alive' signal continuously from both paths (A and B). Once a failure occurs on one path, the source will not receive an acknowledgement from this path and will then switch the traffic to another path.

Let us suppose that the source node has 100 packets to send to the destination node. The source node will select two paths and send 50 packets on each path to the destination node. In a probabilistic scenario in which one link has failed on the network, the source node will resend only 50 packets or less rather than resending all 100 packets as in retransmission.

In this scheme, the proposed framework finds the two energy-efficient routes for the traffic between the embedded nodes, namely the primary and the secondary routes. The main difference between this splitting scheme and the former schemes is the flow conservation constraints in (6.17) and (6.22).

$$\sum_{f \in PN_e} P_{cdef}^{ROUTE1} - \sum_{f \in PN_e} P_{cdfe}^{ROUTE1} \begin{cases} 0.5 \cdot P_{cd}^{TRFP} & if e = c \\ -0.5 \cdot P_{cd}^{TRFP} & if\ e = d \\ 0 & otherwise \end{cases} \quad (6.35)$$

$$\forall\, c, d, e \in PN: c \neq d \text{ and } e \neq f$$

$$\sum_{f \in PN_e} P_{cdef}^{ROUTE2} - \sum_{f \in PN_e} P_{cdfe}^{ROUTE2} \begin{cases} 0.5 \cdot P_{cd}^{TRFP} & if e = c \\ -0.5 \cdot P_{cd}^{TRFP} & if\ e = d \\ 0 & otherwise \end{cases} \quad (6.36)$$

$$\forall\, c, d, e \in PN: c \neq d \text{ and } e \neq f$$

Constraints (6.35) and (6.36) represent the flow conservation constraints for the primary and secondary paths for the traffic splitting scheme.

6.4 Results and Evaluation

To evaluate the performance of the proposed model, we considered a smart building scheme (i.e. enterprise or university campus) where the physical layer is composed of 30 IoT nodes connected by 89 bidirectional wireless links. These IoT nodes are distributed across a campus within an area of 500 m × 500 m as described in Chapter 4. We evaluated the power consumption and the mean traffic latency resulting from the resilient service embedding across distinct zones with the coexistence constraint. The model considered the objective function discussed in Section 6.3.2 for energy efficient-low latency service embedding. The probabilistic model is based on k-connected nodes with the assumption that the network has the ability of failure recovery in the case of a link or node failure. We run our model for two resilience schemes:

6.4.1 Energy-efficient low-latency node-resilient service embedding

For the node-resilient scheme, we run three resilience levels with the objective of minimising the total power consumption and the mean traffic latency:

- Coexistence constraint node resilience (CCNR)
- Partial redundancy node resilience (PRNR)
- Full redundancy node resilience (FRNR)

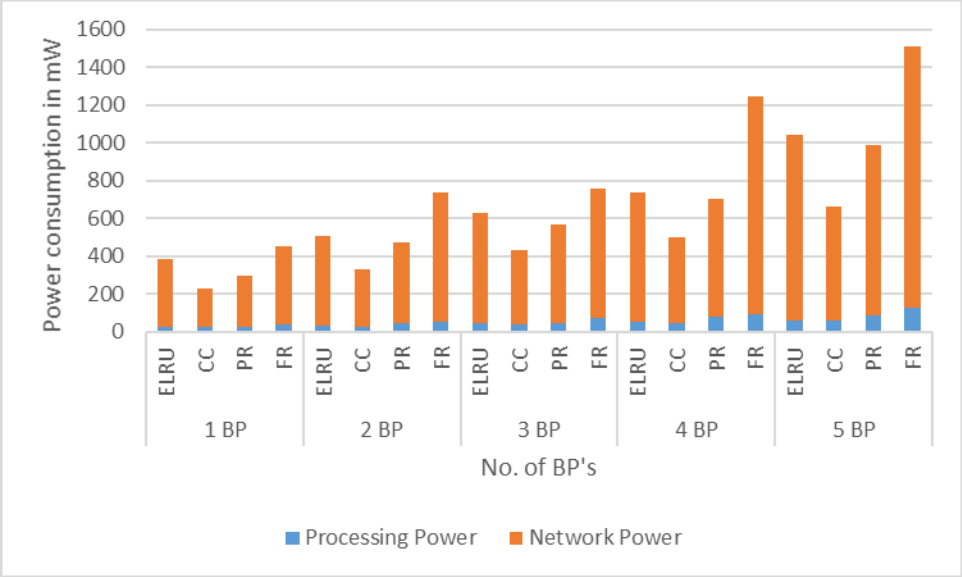


Figure 6-2: Power consumption of energy-efficient low-latency node-resilient service embedding.

The results shown in Figure 6-2 display the total power consumption of CCNR, PRNR, and FRNR and compare them with the energy-latency-resilience unaware (ELRU) scenario. These results demonstrate that the CCNR scenario has an average power saving of 35% compared with the ELRU scenario. While the higher level of power consumption in the PRNR scenario has an average power saving of 10% compared with ELRU.

The FRNR has higher power consumption than the other scenarios, and the average power consumption is 40% higher than that in the ELRU scenario.

The increase in power consumption for each scenario is due to the embedding of the redundant nodes and the traffic among these nodes, but the node resilience level

is improved and the IoT network has the ability to maintain the service provisioning even with a failure in one node.

6.4.2 Energy-efficient low-latency traffic-resilient service embedding

For the traffic-resilient scheme, we run three resilience levels with the objective of minimising the total power consumption and the traffic mean latency:

- Redundancy-based traffic resilience (RDTR)
- Replication-based traffic resilience (RPTR)
- Splitting-based traffic resilience (STR)

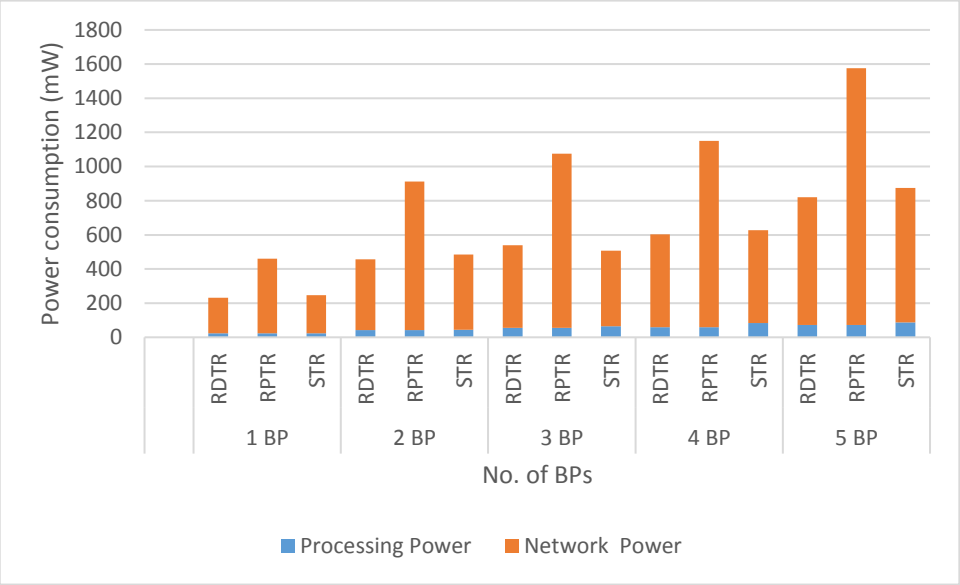


Figure 6-3: Power consumption of traffic-resilient service embedding scenarios without failure.

The results presented in Figure 6-3 display the power consumption of the traffic-resilient service embedding for the RDTR, RPTR, and STR scenarios in the packet delivery case without a failure. These results demonstrated that RDTR has the lowest power consumption and an average power saving of 47% and 4% compared with RPTR and STR scenarios, respectively. Notice than in some cases (i.e. 3 BP's embedding), the STR has consumed led power consumption compared with RDTR, that refers to ability to find energy efficient route for part of traffic (i.e. 50 % of the total traffic) more than the whole traffic in one energy efficient route.

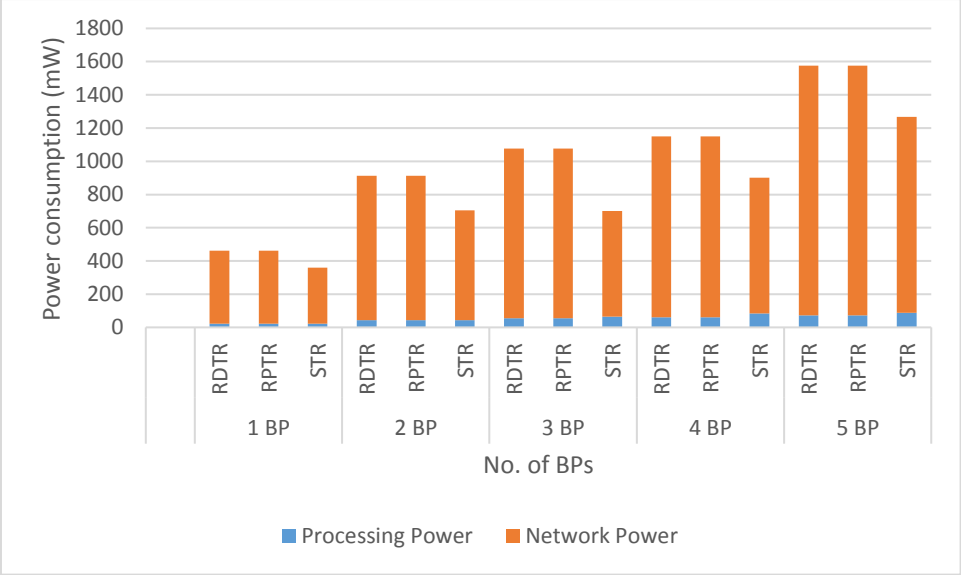


Figure 6-4: Power consumption of traffic-resilient service embedding scenarios with failure.

The results presented in Figure 6-4 display the power consumption of the traffic-resilient service embedding for the RDTR, RPTR, and STR scenarios in the packet delivery case with one link failure. These results demonstrate that RDTR has the same power consumption as RPTR because of the data retransmission through the secondary path. The results also reveal that the STR has an average power saving of 25% compared with the RDTR scenario.

These results show that the proposed technique in the STR scenario has higher power consumption in a successful data delivery by 4% but 25% power savings in the case of one link failure.

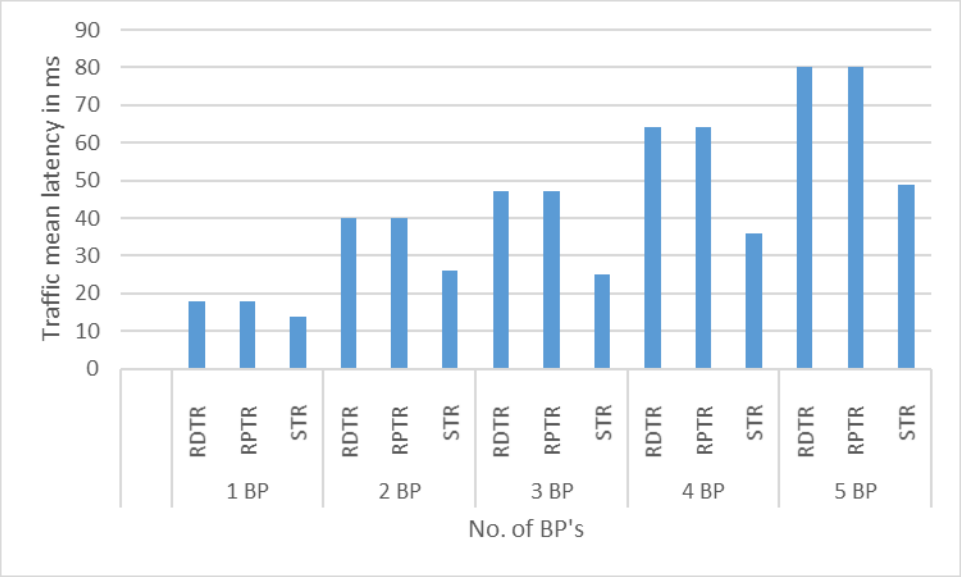


Figure 6-5: Traffic mean latency of traffic resilient service embedding scenarios without failure.

The results presented in Figure 6-5 display the mean network traffic latency of the service embedding scenarios. These results demonstrate that the STR reduces the average mean traffic latency by 37% compared with the RDTR and RPTR scenarios. The mean traffic latency minimisation in STR is due to the traffic splitting and hence the reduction in the arrival rate of the individual nodes. The traffic splitting technique offered better performance in terms of the end-to-end delay.

The packet delivery ratio (PDR) reflects the network performance level, where better network performance resulted in a high packet delivery ratio. The packet delivery ratio is inversely proportional to the network size in IoT networks because the routing performance is better in a low-node-density network.

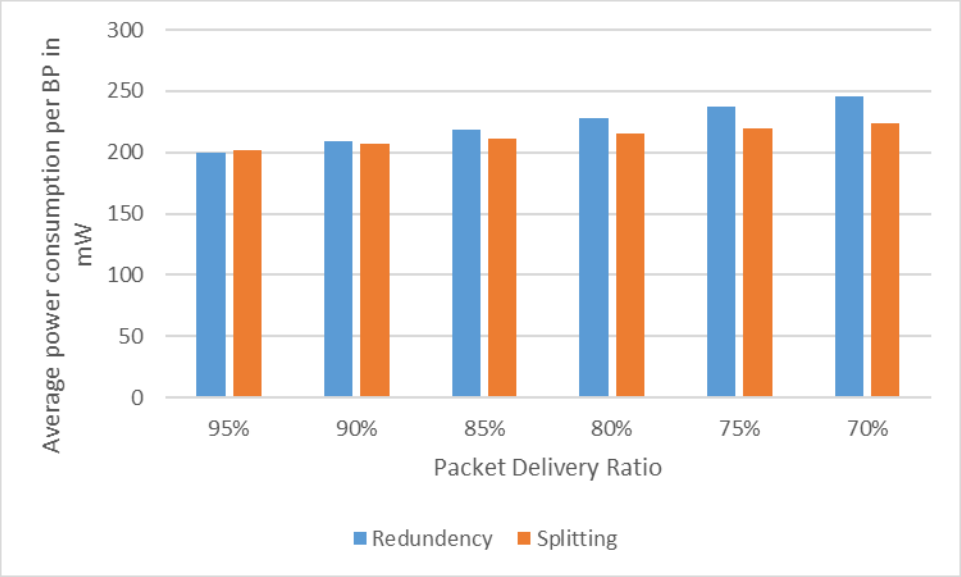


Figure 6-6: Power consumption of traffic-resilient service embedding scenarios for different PDR scenarios.

The results shown in Figure 6-6 present a comparison of the total power consumption in the RDTR and STR scenarios for different PDR values[237]. These results demonstrate that the RDTR is an energy-efficient technique for high-performance networks (i.e. PDR > 95%). However, the STR scenario produces higher power savings with lower PDR. The STR scenario exhibits power savings of 10% compared with RDTR when PDR = 70%. These results help to compare the RDTR and the STR without the RPTR, where the RPTR has the highest power consumption in all the cases.

6.5 Summary

In this chapter, multilevel node and traffic resilience schemes for IoT networks were reviewed. A MILP model to enhance the resilience of services was developed. The node and traffic resilience were enhanced by using the proposed scheme and a model for an energy-efficient low-latency resilient service embedded in a smart building was developed. A range of node and traffic resilience levels were developed and their performance in terms of the mean traffic latency and power consumption were compared. A novel technique was also proposed based on traffic splitting to enhance the network resilience and performance by reducing the packet delivery time. Moreover, splitting techniques were evaluated using redundancy and

replication resilience techniques in terms of the total power consumption and the mean traffic latency for different values of PDR.

The results showed that the STR scenario produced higher power savings with lower PDR: the STR scenario exhibited a power saving of 10% compared with the recent scheme of RDTR when PDR was equal 70%. The results also revealed that the STR reduced the average mean traffic latency by 37% compared with the RDTR and RPTR scenarios. The mean traffic latency minimisation in STR referred to the traffic splitting and reduced the arrival rate of the nodes. The traffic splitting technique also exhibited better performance in terms of the end-to-end delay.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This section summarises the work that has been performed in the present thesis and states its original contributions. The IoT reveals a new era of internet of the future. Many services can be realised by having relevant standard architecture that fulfils the integration with the Internet. Despite great efforts having been made in the progress of standards, more is needed, especially in the areas of architecture and communications. In addition to all the predicted features which make up the IoT, there are several challenges that need to be addressed. Energy efficiency is one of the most important challenges in terms of general concerns and is a significant issue related to the IoT. The IoT is composed of a huge number of nodes that can be deployed indoors or outdoors to monitor relevant factors in their surrounding environments; most of these nodes are battery-powered, and this power limitation means that the IoT is liable to failure. Such failures cause service disruption and require frequent network maintenance, which raises the operational cost. To conquer this issue, researchers and stakeholders have put a great deal of effort into designing energy-efficient solutions to prolong the IoT network lifetime and enhance the performance of the IoT network.

In this thesis, the aim was to contribute significantly to the advancement of IoT technology and to propose a viable architecture for service embedding in IoT networks that prolongs the lifetime of nodes while achieving the required services efficiently and which also enhances the QoS due to reducing data delivery time, and, finally, boosts the resilience of the services.

In Chapter 3, reviewed the current, known proposed architectures for the service virtualisation in IoT networks. We identified the performance metrics and cost. Most of the architectures, protocols and experimental work for service virtualisation architectures have been surveyed in Chapter 3.

In Chapter 4 we evaluated the energy consumption and latency of service embedding in a smart building. We developed a MILP model to embed a virtual topology in the form of BPs into the physical IoT network. Our developed model has three objective functions. The first objective is to minimise the network and processing power consumption. The results of the service embedding in the same zone without coexistence constraint illustrate higher power saving of 63% and 58% for the re-provisional and sequential embedding respectively. In contrast, service embedding in the same zone with coexistence constraint illustrate lower power saving of 36% and 29% for the re-provisional and sequential embedding respectively.

For the service embedding scenario across different zones without coexistence constraint, the results illustrate average power saving of 42% and 22% for the re-provisional and sequential embedding respectively. In contrast, the same embedding with coexistence constraint displays lower power saving of 34% and 17% for the re-provisional and sequential embedding respectively.

The second objective is to minimise the traffic latency in the network; the results of the service embedding across different zones without coexistence constraint illustrate average traffic latency minimisation of 47% and 20% for the re-provisional and sequential embedding respectively. In contrast, the same scenario with coexistence constraint displays average traffic latency minimisation of 34% and 19% for the re-provisional and sequential embedding respectively. In all cases compared with the ELUSE scenario.

The energy-efficient service embedding has highly utilised the energy-efficient routes in the physical network, which leads to higher arrival rates on the energy efficient nodes and consequently increases the traffic latency of these nodes. The traffic latency is an important performance parameter that reduces data delivery time and keeps the traffic in the network within normal utilisation.

The third objective is to jointly minimise the power consumption and traffic latency. We investigated using weight values in the objective function the interplay between power and delay minimisation; and evaluated their impact on the optimality of power saving and traffic latency minimisation for service embedding across different zones with coexistence constraint.

In Chapter 5, we evaluated the power consumption of service embedding in a smart city setting. We proposed a physical IoT network integrated with cloud and fog as processing resources. We developed a MILP model to minimise the network and processing power consumption of the service embedding across different zones with and without coexistence constraint. In addition, we evaluated the processing and network power consumption with different processing latency demands. The results illustrate the highest power consumption with the lowest processing latency constraint. This is due to the embedding of the processing demand in the cloud to satisfy the lowest processing latency constraint. The embedding in the cloud results in higher processing due to the PUE value and traffic power consumption of the core network equipment.

We also investigated the impact of processing splitting on the power consumption of service embedding. The results show that service embedding with processing splitting has power saving of 18% compared with service embedding without the processing splitting scenario under the same processing latency constraint. In addition, we investigated the impact of coexistence constraint on the power consumption of service embedding. The results illustrate that service embedding with processing splitting and without coexistence constraint has an average power saving of 48% compared with service embedding without processing splitting and with coexistence constraint.

In Chapter 6 we enhanced the node and traffic resilience for the proposed architecture in Chapter 4. In chapter 6, we developed a model for an energy-efficient low latency resilient service embedding in a smart building. We evaluated a range of node and traffic resilience levels and compared their performance and cost in terms of traffic mean latency and power consumption. We also proposed a novel technique based on traffic splitting to enhance the network resilience and performance by reducing the packet delivery time. We evaluated splitting techniques using redundancy and replication resilience techniques in terms of total power consumption and traffic mean latency under different values of PDR.

7.2 Future Work

In this section, we propose several possible directions for the topic of service embedding in IoT networks.

7.2.1 Allocation of variable physical resources

Our framework can be extended to allocate suitable resources (i.e. the IoT node) according to the virtual demands in the zones. This kind of optimisation will reduce the number of unused nodes and functions.

7.2.2 High reliability and availability in service embedding

Our framework can be extended to consider more performance metrics, such as node/links reliability and availability. This scheme can be studied by classifying the resources based on the reliability and availability into different levels and the model can either optimise the embedding or find an alternative solution to enhance the network performance.

7.2.3 Prioritised service embedding

Our framework can be extended to add another dimension for the service request, which presents the service priority to be embedded in the highest performance metrics or embedding without blockage.

7.2.4 Energy-efficient solutions

Our framework can be extended to add more parameters to the physical resources, such as remaining battery power, transmission range, and a number of neighbours. These parameters can be considered in our model to extend node lifetime.

7.2.5 Minimising the model complexity

One of the obstacles when using MILP is the processing overhead and requirements. As such, one future direction is to extend the consideration of the virtual demand and physical network by using other optimisation tools.

List of Abbreviations

3GPP	3rd Generation Partnership Project
ALU	Application Level User
ARAT	Active reader active tag
ARPT	Active reader passive tag
BP	Business Process
CAPEX	Capital expenditure
CCNR	Coexistence constraint node resilience
CH	Cluster Head
CS	Compressive Sensing
CSS	Cooperative spectrum sensing
CVO	Composite virtual object
DoS	Denial of Service
DRLS	Desired reliability level scheme
DRX	Discontinuous reception
DTX	Discontinuous transmission
ERDT	Energy-efficient reliable decision transmission
ELUSE	Energy-latency unaware service embedding
FPGA	Field Programmable Gate Arrays
FRNR	Full redundancy node resilience
GSM	Global System for Mobile Communications
HTTP	Hyper Text Transfer Protocol
IaaS	Infrastructure as a service
ICT	The information and Communication technology
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
InP	Infrastructure Provider
IoT	Internet of Things
ITU	International Telecommunication Union
LTE	Long-Term Evolution
M2M	Machine to machine
MAC	Media access control
MILP	mixed integer linear programming
MIT	Massachusetts Institute of Technology
NaaS	Network as a Service
NFC	Near-field communication
OPEX	Operation Expenditure
PaaS	Platform as a service
PRAT	passive reader active tag
PRNR	Partial redundancy node resilience
PRT	Packet Replication Techniques
QoS	Quality of Service
QSI	Quick Sleeping Indication

RAN	Radio Access Network
RBDS	Reliability-based data compression scheme
RBS	Reliability-based sub-channel scheme
REST	Representational State Transfer
RESE	Real time energy efficient service embedding
RDTR	Redundancy-based traffic resilience
RLSE	Real time low latency service embedding
RFID	Radio-frequency identification
ROA	Resource Oriented Architecture
RPL	Routing Protocol for Low-Power and Lossy Networks
RPTR	Replication-based traffic resilience
RWO	Real world object
SaaS	Software as a service
SBRS	Standby route selection scheme
SD	Service discovery
SDN	Software Defined Network
SenaaS	Sensor as a Service
SOA	Service Oriented Architecture
SOC	system-on-chip
SP	Service Provider
STR	Splitting-based traffic resilience
URI	Uniform resource identifiers
VM	Virtual Machine
VO	Virtual object
WoT	Web of Things
WSN	Wireless Sensor network

List of References

[1] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. J. I. N. Polakos, "Wireless sensor network virtualization: early architecture and research perspectives," vol. 29, no. 3, pp. 104-112, 2015.

[2] A. N. Al-Quzweeni, A. Q. Lawey, T. E. Elgorashi, and J. M. Elmirghani, "Optimized Energy Aware 5G Network Function Virtualization," 2018.

[3] D. Giusto, A. Iera, G. Morabito, and L. Atzori, *The internet of things: 20th Tyrrhenian workshop on digital communications*. Springer Science & Business Media, 2010.

[4] S. Madakam, R. Ramaswamy, S. J. J. o. C. Tripathi, and Communications, "Internet of Things (IoT): A literature review," vol. 3, no. 05, p. 164, 2015.

[5] Z. Jun, D. Simplot-Ryl, C. Bisdikian, and H. J. I. C. M. Mouftah, "The internet of things," vol. 49, no. 11, pp. 30-31, 2011.

[6] M. P. Michael and M. Darianian, "Architectural solutions for mobile RFID services for the internet of things," in *2008 IEEE Congress on Services-Part I*, 2008: IEEE, pp. 71-74.

[7] F. Wortmann and K. Flüchter, "Internet of things," *Business & Information Systems Engineering,* vol. 57, no. 3, pp. 221-224, 2015.

[8] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IOT) technologies, applications and challenges," in *2016 IEEE Smart Energy Grid Engineering (SEGE)*, 2016: IEEE, pp. 381-385.

[9] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express,* vol. 5, no. 1, pp. 1-7, 2019.

[10] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A survey on facilities for experimental internet of things research," *IEEE Communications Magazine,* vol. 49, no. 11, 2011.

[11] H. Ning and H. Liu, "Cyber-physical-social based security architecture for future internet of things," *Advances in Internet of Things,* vol. 2, no. 01, p. 1, 2012.

[12] R. G. Helps and S. J. Pack, "Cyber-physical system concepts for IT students," in *Proceedings of the 14th annual*

ACM SIGITE conference on Information technology education, 2013: ACM, pp. 7-12.

[13] V. Cozzolino, A. Y. Ding, A. A. Sani, R. M. Mortier, D. Kutscher, and J. Ott, "Empowering Cyber-Physical Systems with FADEX," 2018.

[14] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," *IEEE Internet Computing,* vol. 14, no. 1, pp. 44-51, 2009.

[15] M. R. F. B. Junior, C. L. Batista, M. E. Marques, and C. R. M. Pessoa, "Business Models Applicable to IoT," in *Handbook of Research on Business Models in Modern Competitive Scenarios*: IGI Global, 2019, pp. 21-42.

[16] K. Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010.

[17] J. Liu and W. Tong, "Adaptive service framework based on grey decision-making in the internet of things," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010: IEEE, pp. 1-4.

[18] L. M. Camarinha-Matos, S. Tomic, and P. Graça, *Technological Innovation for the Internet of Things: 4th IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2013, Costa de Caparica, Portugal, April 15-17, 2013, Proceedings*. Springer, 2013.

[19] K. T. Nguyen, M. Laurent, and N. J. A. H. N. Oualha, "Survey on secure communication protocols for the Internet of Things," vol. 32, pp. 17-31, 2015.

[20] K. Zheng, F. Hu, W. Xiang, M. Dohler, and W. J. a. p. a. Wang, "Radio resource allocation in LTE-advanced cellular networks with M2M communications," 2015.

[21] F. Xia, Y.-C. Tian, Y. Li, and Y. J. S. Sung, "Wireless sensor/actuator network design for mobile control applications," vol. 7, no. 10, pp. 2157-2173, 2007.

[22] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. J. I. C. m. Cayirci, "A survey on sensor networks," vol. 40, no. 8, pp. 102-114, 2002.

[23] M. Bacco, A. Gotta, C. Roseti, and F. Zampognaro, "A study on TCP error recovery interaction with random access satellite schemes," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal*

Processing for Space Communications Workshop (ASMS/SPSC), 2014: IEEE, pp. 405-410.

[24] N. Omnes, M. Bouillon, G. Fromentoux, and O. Le Grand, "A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges," in *2015 18th International Conference on Intelligence in Next Generation Networks*, 2015: IEEE, pp. 64-69.

[25] D. Niyato, X. Lu, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," 2011.

[26] S. Bhardwaj, L. Jain, S. J. I. J. o. e. Jain, and i. Technology, "Cloud computing: A study of infrastructure as a service (IAAS)," vol. 2, no. 1, pp. 60-63, 2010.

[27] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges," in *2010 24th IEEE international conference on advanced information networking and applications*, 2010: Ieee, pp. 27-33.

[28] M. A. J. C. A. Cusumano, "Cloud computing and SaaS as new computing platforms," vol. 53, no. 4, pp. 27-29, 2010.

[29] C. Perera, D. S. Talagala, C. H. Liu, and J. C. Estrella, "Energy-Efficient Location and Activity-Aware On-Demand Mobile Distributed Sensing Platform for Sensing as a Service in IoT Clouds," *IEEE Transactions on Computational Social Systems,* vol. 2, no. 4, pp. 171-181, 2015.

[30] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014: IEEE, pp. 375-376.

[31] J. Gubbi, R. Buyya, S. Marusic, and M. J. F. g. c. s. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," vol. 29, no. 7, pp. 1645-1660, 2013.

[32] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. J. I. N. Ott, "Consolidate IoT edge computing with lightweight virtualization," vol. 32, no. 1, pp. 102-111, 2018.

[33] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *2014 International Conference on Future Internet of Things and Cloud*, 2014: IEEE, pp. 464-470.

[34] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, "Greening iot with fog: A survey," in *Edge Computing (EDGE), 2017 IEEE International Conference on*, 2017: IEEE, pp. 25-31.

[35] C. Thota, R. Sundarasekar, G. Manogaran, R. Varatharajan, and M. Priyan, "Centralized fog computing security platform for IoT and cloud in healthcare system," in *Fog Computing: Breakthroughs in Research and Practice*: IGI Global, 2018, pp. 365-378.

[36] A. A. Mutlag, M. K. A. Ghani, N. Arunkumar, M. A. Mohamed, and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," *Future Generation Computer Systems,* vol. 90, pp. 62-78, 2019.

[37] S. H. Mohamed, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficiency of server-centric PON data center architecture for fog computing," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*, 2018: IEEE, pp. 1-4.

[38] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering,* vol. 2017, 2017.

[39] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *2012 10th international conference on frontiers of information technology*, 2012: IEEE, pp. 257-260.

[40] A. Gaur, B. Scotney, G. Parr, and S. J. P. c. s. McClean, "Smart city architecture and its applications based on IoT," vol. 52, pp. 1089-1094, 2015.

[41] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. J. I. I. o. T. J. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," vol. 4, no. 5, pp. 1125-1142, 2017.

[42] M. R. Palattella *et al.*, "Internet of things in the 5G era: Enablers, architecture, and business models," vol. 34, no. 3, pp. 510-527, 2016.

[43] S. Dhingra, R. B. Madda, A. H. Gandomi, R. Patan, and M. Daneshmand, "Internet of Things Mobile-Air Pollution Monitoring System (IoT-Mobair)," *IEEE Internet of Things Journal,* 2019.

[44] M. Ahmad, T. Younis, M. A. Habib, R. Ashraf, and S. H. Ahmed, "A Review of Current Security Issues in Internet of Things," in *Recent Trends and Advances in Wireless and IoT-enabled Networks*: Springer, 2019, pp. 11-23.

[45] P. Evensen and H. Meling, "SenseWrap: A service oriented middleware with sensor virtualization and self-configuration," in *2009 International Conference on*

Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009: IEEE, pp. 261-266.

[46] Y. Hong, "A resource-oriented middleware framework for heterogeneous internet of things," in *2012 International Conference on Cloud and Service Computing*, 2012: IEEE, pp. 12-16.

[47] Z. Khalid, N. Fisal, and M. J. S. Rozaini, "A survey of middleware for sensor and network virtualization," vol. 14, no. 12, pp. 24046-24097, 2014.

[48] W. Zhiliang, Y. Yi, W. Lu, and W. Wei, "A SOA based IOT communication middleware," in *Mechatronic science, electric engineering and computer (MEC), 2011 International conference on*, 2011: IEEE, pp. 2555-2558.

[49] S. Ezdiani, I. S. Acharyya, S. Sivakumar, and A. Al-Anbuky, "An IoT environment for WSN adaptive QoS," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, 2015: IEEE, pp. 586-593.

[50] O. Othman and D. C. Schmidt, "Issues in the design of adaptive middleware load balancing," in *ACM SIGPLAN Notices*, 2001, vol. 36, no. 8: ACM, pp. 205-213.

[51] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials,* vol. 17, no. 4, pp. 2347-2376, 2015.

[52] T. Yashiro, S. Kobayashi, N. Koshizuka, and K. Sakamura, "An Internet of Things (IoT) architecture for embedded appliances," in *2013 IEEE Region 10 Humanitarian Technology Conference*, 2013: IEEE, pp. 314-319.

[53] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications,* vol. 54, pp. 1-31, 2014.

[54] J. W. Hui and D. E. Culler, "IPv6 in low-power wireless networks," *Proceedings of the IEEE,* vol. 98, no. 11, pp. 1865-1878, 2010.

[55] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems,* vol. 29, no. 7, pp. 1645-1660, 2013.

[56] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017: IEEE, pp. 492-496.

[57] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. J. A. H. N. Agrawal, "Choices for interaction with things on Internet and underlying issues," vol. 28, pp. 68-90, 2015.

[58] A. F. Molisch *et al.*, "IEEE 802.15. 4a channel model-final report," *IEEE P802,* vol. 15, no. 04, p. 0662, 2004.

[59] L.-H. Yen and W.-T. Tsai, "The room shortage problem of tree-based ZigBee/IEEE 802.15. 4 wireless networks," *Computer Communications,* vol. 33, no. 4, pp. 454-462, 2010.

[60] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: next-generation wireless broadband technology," *IEEE wireless communications,* vol. 17, no. 3, pp. 10-22, 2010.

[61] J. Liu, H. Guo, H. Nishiyama, H. Ujikawa, K. Suzuki, and N. Kato, "New perspectives on future smart FiWi networks: scalability, reliability, and energy efficiency," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 2, pp. 1045-1072, 2016.

[62] I. Farris *et al.*, "Social virtual objects in the edge cloud," vol. 2, no. 6, pp. 20-28, 2015.

[63] M. O. Musa, T. E. El-Gorashi, and J. M. Elmirghani, "Bounds on GreenTouch GreenMeter Network Energy Efficiency," *Journal of Lightwave Technology,* vol. 36, no. 23, pp. 5395-5405, 2018.

[64] J. Huang, Y. Meng, X. Gong, Y. Liu, and Q. Duan, "A novel deployment scheme for green internet of things," *IEEE Internet of Things Journal,* vol. 1, no. 2, pp. 196-205, 2014.

[65] A. Atrey, N. Jain, and N. Iyengar, "A study on green cloud computing," *International Journal of Grid and Distributed Computing,* vol. 6, no. 6, pp. 93-10, 2013.

[66] D. Cavdar and F. Alagoz, "A survey of research on greening data centers," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012: IEEE, pp. 3237-3242.

[67] S. F. Abedin, M. G. R. Alam, R. Haw, and C. S. Hong, "A system model for energy efficient green-IoT network," in *2015 International Conference on Information Networking (ICOIN)*, 2015, pp. 177-182.

[68] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: early progress and back to the future," *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 8, no. 1, pp. 1-21, 2012.

[69] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks,* vol. 54, no. 15, pp. 2787-2805, 2010.

[70] A. Yachir, Y. Amirat, A. Chibani, and N. Badache, "Event-aware framework for dynamic services discovery and selection in the context of ambient intelligence and Internet of Things," *IEEE Transactions on Automation Science and Engineering,* vol. 13, no. 1, pp. 85-102, 2016.

[71] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Communications Magazine,* vol. 48, no. 6, pp. 92-101, 2010.

[72] D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," in *2015 global conference on communication technologies (GCCT)*, 2015: IEEE, pp. 169-173.

[73] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, "IoT based smart security and home automation system," in *2016 international conference on computing, communication and automation (ICCCA)*, 2016: IEEE, pp. 1286-1289.

[74] K. Mandula, R. Parupalli, C. A. Murty, E. Magesh, and R. Lunagariya, "Mobile based home automation using Internet of Things (IoT)," in *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2015: IEEE, pp. 340-343.

[75] S. Pirbhulal *et al.*, "A novel secure IoT-based smart home automation system using a wireless sensor network," *Sensors,* vol. 17, no. 1, p. 69, 2017.

[76] V. Patchava, H. B. Kandala, and P. R. Babu, "A smart home automation technique with raspberry pi using iot," in *2015 International Conference on Smart Sensors and Systems (IC-SSS)*, 2015: IEEE, pp. 1-4.

[77] E. Avilés-López and J. A. García-Macías, "Mashing up the Internet of Things: a framework for smart environments," *EURASIP Journal on Wireless Communications and Networking,* vol. 2012, no. 1, p. 79, 2012.

[78] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE Journal on Selected Areas in Communications,* vol. 34, no. 12, pp. 4077-4090, 2016.

[79] D. Cook and S. K. Das, *Smart environments: Technology, protocols and applications*. John Wiley & Sons, 2004.

[80] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Generation Computer Systems,* vol. 78, pp. 659-676, 2018.

[81] C. Doukas and I. Maglogiannis, "Bringing IoT and cloud computing towards pervasive healthcare," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2012: IEEE, pp. 922-926.

[82] L. Catarinucci *et al.*, "An IoT-aware architecture for smart healthcare systems," *IEEE Internet of Things Journal,* vol. 2, no. 6, pp. 515-526, 2015.

[83] T.-Y. Kim, S. Youm, J.-J. Jung, and E.-J. Kim, "Multi-hop WBAN construction for healthcare IoT systems," in *2015 International Conference on Platform Technology and Service*, 2015: IEEE, pp. 27-28.

[84] F. Fernandez and G. C. Pallis, "Opportunities and challenges of the Internet of Things for healthcare: Systems engineering perspective," in *2014 4th International Conference on Wireless Mobile Communication and Healthcare-Transforming Healthcare Through Innovations in Mobile and Wireless Technologies (MOBIHEALTH)*, 2014: IEEE, pp. 263-266.

[85] N. Luo, W. Zhong, F. Wan, Z. Ye, and F. Qian, "An agent-based service-oriented integration architecture for chemical process automation," *Chinese Journal of Chemical Engineering,* vol. 23, no. 1, pp. 173-180, 2015.

[86] W. Shen, Q. Hao, S. Wang, Y. Li, and H. Ghenniwa, "An agent-based service-oriented integration architecture for collaborative intelligent manufacturing," *Robotics and Computer-Integrated Manufacturing,* vol. 23, no. 3, pp. 315-325, 2007.

[87] Z. D. R. Gnimpieba, A. Nait-Sidi-Moh, D. Durand, and J. J. P. C. S. Fortin, "Using Internet of Things technologies for a collaborative supply chain: Application to tracking of pallets and containers," vol. 56, pp. 550-557, 2015.

[88] P. Ferreira, R. Martinho, and D. Domingos, "IoT-aware business processes for logistics: limitations of current approaches," in *Inforum*, 2010, vol. 3, no. 2010, pp. 612-613.

[89] W. Liu and Z. Gao, "Study on IOT based architecture of logistics service supply," 2014.

[90] D. Kyriazis, T. Varvarigou, D. White, A. Rossi, and J. Cooper, "Sustainable smart city IoT applications: Heat and electricity management & Eco-conscious cruise control for public transportation," in *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, 2013: IEEE, pp. 1-5.

[91] T. M. Bojan, U. R. Kumar, and V. M. Bojan, "An internet of things based intelligent transportation system," in *2014 IEEE International Conference on Vehicular Electronics and Safety*, 2014: IEEE, pp. 174-179.

[92] W. He, G. Yan, and L. Da Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics,* vol. 10, no. 2, pp. 1587-1595, 2014.

[93] J.-M. Liang, J.-J. Chen, H.-H. Cheng, and Y.-C. Tseng, "An energy-efficient sleep scheduling with QoS consideration in 3GPP LTE-advanced networks for internet of things," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 3, no. 1, pp. 13-22, 2013.

[94] M. Jin, S. Gao, H. Luo, J. Li, Y. Zhang, and S. K. Das, "An Approach to Pre-Schedule Traffic in Time-Dependent Pricing Systems," *IEEE Transactions on Network and Service Management,* vol. 16, no. 1, pp. 334-347, 2019.

[95] L. Tan and N. Wang, "Future internet: The internet of things," in *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, 2010, vol. 5: IEEE, pp. V5-376-V5-380.

[96] G. Tselentis, J. Domingue, and A. Galis, *Towards the future internet: A European research perspective*. IOS press, 2009.

[97] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between internet of things and social networks: Review and research challenges," *IEEE Internet of Things Journal,* vol. 1, no. 3, pp. 206-215, 2014.

[98] R. Liu and I. J. Wassell, "Opportunities and challenges of wireless sensor networks using cloud services," in *Proceedings of the workshop on Internet of Things and Service Platforms*, 2011: ACM, p. 4.

[99] R. Chen, J. Guo, and F. Bao, "Trust management for service composition in SOA-based IoT systems," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, 2014: IEEE, pp. 3444-3449.

[100] R. Chen, J. Guo, and F. J. I. T. o. S. C. Bao, "Trust management for SOA-based IoT and its application to service composition," vol. 9, no. 3, pp. 482-495, 2016.
[101] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002, vol. 3: IEEE, pp. 1567-1576.
[102] M. Z. Jacobson, "Review of solutions to global warming, air pollution, and energy security," *Energy & Environmental Science,* vol. 2, no. 2, pp. 148-173, 2009.
[103] M. Moreno, B. Úbeda, A. Skarmeta, and M. Zamora, "How can we tackle energy efficiency in iot basedsmart buildings?," *Sensors,* vol. 14, no. 6, pp. 9582-9614, 2014.
[104] Y. Lim, S. Daas, S. Hashim, R. Sidek, N. Kamsani, and F. Rokhani, "Reduced hardware architecture for energy-efficient IoT healthcare sensor nodes," in *2015 IEEE International Circuits and Systems Symposium (ICSyS)*, 2015: IEEE, pp. 90-95.
[105] Z. T. Al-Azez, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Virtualization framework for energy efficient IoT networks," in *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, 2015: IEEE, pp. 74-77.
[106] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy Efficient and Reliable Transport of Data in Cloud-Based IoT," *IEEE Access,* vol. 7, pp. 64641-64650, 2019.
[107] M. E. Khanouche, Y. Amirat, A. Chibani, M. Kerkar, and A. Yachir, "Energy-centered and QoS-aware services selection for Internet of Things," *IEEE Transactions on Automation Science and Engineering,* vol. 13, no. 3, pp. 1256-1269, 2016.
[108] P. Spiess *et al.*, "SOA-based integration of the internet of things in enterprise services," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009: IEEE, pp. 968-975.
[109] S. Clement, D. McKee, and J. Xu, "Service-Oriented Reference Architecture for Smart Cities," in *Service-Oriented System Engineering (SOSE), 2017 IEEE Symposium on*, 2017: IEEE, pp. 81-85.
[110] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks,* vol. 67, pp. 104-122, 2014.
[111] S. Abdullah and K. Yang, "A QoS aware message scheduling algorithm in Internet of Things environment," in

2013 IEEE Online Conference on Green Communications (OnlineGreenComm), 2013: IEEE, pp. 175-180.

[112] I. Al-Anbagi, M. Erol-Kantarci, H. T. J. I. C. S. Mouftah, and Tutorials, "A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications," vol. 18, no. 1, pp. 525-552, 2016.

[113] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE network,* vol. 31, no. 1, pp. 52-58, 2017.

[114] J. M. Llopis, J. Pieczerak, and T. Janaszka, "Minimizing latency of critical traffic through SDN," in *2016 IEEE International Conference on Networking, Architecture and Storage (NAS)*, 2016: IEEE, pp. 1-6.

[115] F. Xing and W. Wang, "Analyzing resilience to node misbehaviors in wireless multi-hop networks," in *2007 IEEE Wireless Communications and Networking Conference*, 2007: IEEE, pp. 3489-3494.

[116] K. Benson, "Enabling resilience in the Internet of Things," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015: IEEE, pp. 230-232.

[117] Y. Tsado, D. Lund, and K. Gamage, "Resilient wireless communication networking for Smart grid BAN," in *2014 IEEE International Energy Conference (ENERGYCON)*, 2014: IEEE, pp. 846-851.

[118] Y. Wang, "System resilience quantification for probabilistic design of Internet-of-Things architecture," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016: American Society of Mechanical Engineers, pp. V01BT02A011-V01BT02A011.

[119] K. E. Nolan, M. Y. Kelly, M. Nolan, J. Brady, and W. Guibene, "Techniques for resilient real-world IoT," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016: IEEE, pp. 222-226.

[120] G. Suciu, A. Vulpe, S. Halunga, O. Fratu, G. Todoran, and V. Suciu, "Smart cities built on resilient cloud computing and secure internet of things," in *2013 19th international conference on control systems and computer science*, 2013: IEEE, pp. 513-518.

[121] J. N. Al-Karaki and A. E. J. I. w. c. Kamal, "Routing techniques in wireless sensor networks: a survey," vol. 11, no. 6, pp. 6-28, 2004.

[122] J. P. Sterbenz, "Smart city and IoT resilience, survivability, and disruption tolerance: Challenges, modelling, and a survey of research opportunities," in *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2017: IEEE, pp. 1-6.
[123] Y. Huang, J.-F. Martínez, J. Sendra, and L. J. S. López, "Resilientwireless sensor networks using topology control: a review," vol. 15, no. 10, pp. 24735-24770, 2015.
[124] H. Alwan and A. Agarwal, "A survey on fault tolerant routing techniques in wireless sensor networks," in *2009 Third International Conference on Sensor Technologies and Applications*, 2009: IEEE, pp. 366-371.
[125] S. M. Oteafy and H. S. J. I. I. o. T. J. Hassanein, "Resilient IoT architectures over dynamic sensor networks with adaptive components," vol. 4, no. 2, pp. 474-483, 2017.
[126] R. Dou and G. J. I. S. J. Nan, "Optimizing sensor network coverage and regional connectivity in industrial IoT systems," vol. 11, no. 3, pp. 1351-1360, 2017.
[127] K. Laubhan, K. Talaat, S. Riehl, M. S. Aman, A. Abdelgawad, and K. Yelamarthi, "A low-power IoT framework: From sensors to the cloud," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, 2016: IEEE, pp. 0648-0652.
[128] C. Kiraly, T. Istomin, O. Iova, and G. P. Picco, "D-RPL: Overcoming memory limitations in RPL point-to-multipoint routing," in *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, 2015: IEEE, pp. 157-160.
[129] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *International Journal of Distributed Sensor Networks,* vol. 9, no. 8, p. 794326, 2013.
[130] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *2015 International Conference on Pervasive Computing (ICPC)*, 2015: IEEE, pp. 1-6.
[131] F. J. C. C. Al-Turjman, "QoS—aware data delivery framework for safety-inspired multimedia in integrated vehicular-IoT," vol. 121, pp. 33-43, 2018.
[132] Q. Le, T. Ngo-Quynh, and T. Magedanz, "RPL-based multipath routing protocols for Internet of Things on wireless sensor networks," in *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, 2014: IEEE, pp. 424-429.
[133] N. Maalel, E. Natalizio, A. Bouabdallah, P. Roux, and M. Kellil, "Reliability for emergency applications in internet of

things," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, 2013: IEEE, pp. 361-366.

[134] D. P. Abreu, K. Velasquez, M. Curado, and E. J. A. o. T. Monteiro, "A resilient Internet of Things architecture for smart cities," journal article vol. 72, no. 1, pp. 19-30, February 01 2017.

[135] P. Thubert, M. R. Palattella, and T. Engel, "6TiSCH centralized scheduling: When SDN meet IoT," in *Proc. of IEEE Conf. on Standards for Communications & Networking (CSCN'15)*, 2015.

[136] P. Kedia, R. Nagpal, and T. P. J. I. J. o. C. A. Singh, "A survey on virtualization service providers, security issues, tools and future trends," vol. 69, no. 24, 2013.

[137] N. Lucas Martínez, J.-F. Martínez, and V. J. S. Hernández Díaz, "Virtualization of event sources in wireless sensor networks for the internet of things," vol. 14, no. 12, pp. 22737-22753, 2014.

[138] C. Verdouw, A. Beulens, and J. Van Der Vorst, "Virtualisation of floricultural supply chains: A review from an Internet of Things perspective," *Computers and electronics in agriculture,* vol. 99, pp. 160-175, 2013.

[139] M. M. Islam and E.-N. J. J. o. N. Huh, "Virtualization in wireless sensor network: Challenges and opportunities," vol. 7, no. 3, p. 412, 2012.

[140] H. Ko, J. Jin, and S. L. Keoh, "Secure service virtualization in IoT by dynamic service dependency verification," *IEEE Internet of Things Journal,* vol. 3, no. 6, pp. 1006-1014, 2016.

[141] D. Kelaidonis *et al.*, "Virtualization and cognitive management of real world objects in the internet of things," in *2012 IEEE International Conference on Green Computing and Communications*, 2012: IEEE, pp. 187-194.

[142] A. Celesti, D. Mulfari, M. Fazio, M. Villari, and A. Puliafito, "Exploring container virtualization in IoT clouds," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016: IEEE, pp. 1-6.

[143] M. Islam, M. M. Hassan, G.-W. Lee, and E.-N. J. S. Huh, "A survey on virtualization of wireless sensor networks," vol. 12, no. 2, pp. 2175-2207, 2012.

[144] A. L. Valdivieso Caraguay, A. Benito Peral, L. I. Barona Lopez, and L. J. J. I. J. o. D. S. N. Garcia Villalba, "SDN: Evolution and opportunities in the development IoT applications," vol. 10, no. 5, p. 735142, 2014.

[145] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," vol. 18, no. 1, pp. 236-262, 2016.
[146] C. Liang, F. R. J. I. C. S. Yu, and Tutorials, "Wireless network virtualization: A survey, some research issues and challenges," vol. 17, no. 1, pp. 358-380, 2015.
[147] S. Nastic, S. Sehic, D.-H. Le, H.-L. Truong, and S. Dustdar, "Provisioning software-defined IoT cloud systems," in *2014 international conference on future internet of things and cloud*, 2014: IEEE, pp. 288-295.
[148] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access,* vol. 4, pp. 5591-5606, 2016.
[149] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in *2014 IEEE network operations and management symposium (NOMS)*, 2014: IEEE, pp. 1-9.
[150] H. Li, M. Dong, and K. Ota, "Radio access network virtualization for the social Internet of Things," *IEEE Cloud Computing,* vol. 2, no. 6, pp. 42-50, 2015.
[151] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes, and M. J. I. C. M. Mohammadi, "Toward better horizontal integration among IoT services," vol. 53, no. 9, pp. 72-79, 2015.
[152] A. Mahmud, R. Rahmani, and T. Kanter, "Deployment of flow-sensors in internet of things' virtualization via openflow," in *2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing*, 2012: IEEE, pp. 195-200.
[153] Q. Duan, Y. Yan, A. V. J. I. T. o. N. Vasilakos, and S. Management, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," vol. 9, no. 4, pp. 373-392, 2012.
[154] C.-T. Kuo, P.-W. Chi, V. Chang, and C.-L. J. F. G. C. S. Lei, "SFaaS: Keeping an eye on IoT fusion environment with security fusion as a service," vol. 86, pp. 1424-1436, 2018.
[155] I. Khan *et al.*, "Wireless sensor network virtualization: A survey," vol. 18, no. 1, pp. 553-576, 2016.
[156] Y. Li, X. Su, J. Riekki, T. Kanter, and R. Rahmani, "A SDN-based architecture for horizontal Internet of Things services," in *2016 IEEE International Conference on Communications (ICC)*, 2016: IEEE, pp. 1-7.

[157] H.-L. Truong and S. J. I. C. C. Dustdar, "Principles for engineering IoT cloud systems," vol. 2, no. 2, pp. 68-76, 2015.
[158] P. Vlacheas *et al.*, "Enabling smart cities through a cognitive management framework for the internet of things," vol. 51, no. 6, pp. 102-111, 2013.
[159] A. Meloni, P. A. Pegoraro, L. Atzori, A. Benigni, and S. J. C. N. Sulis, "Cloud-based IoT solution for state estimation in smart grids: Exploiting virtualization and edge-intelligence technologies," vol. 130, pp. 156-165, 2018.
[160] A. Meloni, P. A. Pegoraro, L. Atzori, and S. Sulis, "An IoT architecture for wide area measurement systems: a virtualized PMU based approach," in *2016 IEEE International Energy Conference (ENERGYCON)*, 2016: IEEE, pp. 1-6.
[161] V. Foteinos, D. Kelaidonis, G. Poulios, P. Vlacheas, V. Stavroulaki, and P. J. I. v. t. m. Demestichas, "Cognitive management for the internet of things: A framework for enabling autonomous applications," vol. 8, no. 4, pp. 90-99, 2013.
[162] D. Kelaidonis *et al.*, "A cognitive management framework for smart objects and applications in the internet of things," in *International Conference on Mobile Networks and Management*, 2012: Springer, pp. 196-206.
[163] C. Savaglio and G. Fortino, "Autonomic and cognitive architectures for the Internet of Things," in *International Conference on Internet and Distributed Computing Systems*, 2015: Springer, pp. 39-47.
[164] R. J. I. A. Morabito, "Virtualization on internet of things edge devices with container technologies: a performance evaluation," vol. 5, pp. 8835-8850, 2017.
[165] R. Morabito, J. Kjällman, and M. Komu, "Hypervisors vs. lightweight virtualization: a performance comparison," in *2015 IEEE International Conference on Cloud Engineering*, 2015: IEEE, pp. 386-393.
[166] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues," in *2010 Second International Conference on Computer and Network Technology*, 2010: IEEE, pp. 222-226.
[167] R. Morabito and N. Beijar, "Enabling data processing at the network edge through lightweight virtualization technologies," in *2016 IEEE International Conference on*

Sensing, Communication and Networking (SECON Workshops), 2016: IEEE, pp. 1-6.

[168] D. Mulfari, M. Fazio, A. Celesti, M. Villari, and A. Puliafito, "Design of an IoT cloud system for container virtualization on smart objects," in *European Conference on Service-Oriented and Cloud Computing*, 2015: Springer, pp. 33-47.

[169] M. M. Islam, E.-N. J. U. C. Huh, and M. Applications, "Virtualization of wireless sensor network: Smart house perspective," vol. 7, pp. 54-61, 2012.

[170] F. Berkers, M. Roelands, F. Bomhof, T. Bachet, M. Van Rijn, and W. Koers, "Constructing a multi-sided business model for a smart horizontal IoT service platform," in *2013 17th International Conference on Intelligence in Next Generation Networks (ICIN)*, 2013: IEEE, pp. 126-132.

[171] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE transactions on Services Computing,* vol. 3, no. 3, pp. 223-235, 2010.

[172] K. Dar, A. Taherkordi, R. Rouvoy, and F. Eliassen, "Adaptable service composition for very-large-scale internet of things systems," in *Proceedings of the 8th Middleware Doctoral Symposium*, 2011: ACM, p. 2.

[173] P. Kumar, "Some Observations on Dependency Analysis of SOA Based Systems," *International Journal of Information Technology and Computer Science (IJITCS),* vol. 8, no. 1, p. 54, 2016.

[174] P. Evensen and H. Meling, "Sensor virtualization with self-configuration and flexible interactions," in *Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, 2009: ACM, pp. 31-38.

[175] S. Alam, M. M. Chowdhury, and J. Noll, "Senaas: An event-driven sensor virtualization approach for internet of things cloud," in *2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications*, 2010: IEEE, pp. 1-6.

[176] S. Alam, M. M. Chowdhury, J. J. I. J. o. D. Noll, Analysis, T. f. I. Circuits, and Systems, "Virtualizing sensor for the enablement of semantic-aware internet of things ecosystem," vol. 2, no. 1, p. 41, 2011.

[177] M. J. I. I. J. o. E. E. Kaur, "Advances in embedded service oriented architecture (SOA) for resource-constrained devices," vol. 3, no. 7, pp. 14-8, 2015.

[178] S. Karnouskos, O. Baecker, L. Moreira Sá de Souza, and P. Spiess, "Integration of SOA-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure," 2007.
[179] P. Langendoerfer, K. Piotrowski, M. Diaz, and B. Rubio, "Distributed shared memory as an approach for integrating wsns and cloud computing," in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012: IEEE, pp. 1-6.
[180] A. Madhavapeddy and D. J. J. C. o. t. A. Scott, "Unikernels: the rise of the virtual library operating system," vol. 57, no. 1, pp. 61-69, 2014.
[181] A. Bratterud, A.-A. Walla, H. Haugerud, P. E. Engelstad, and K. Begnum, "IncludeOS: A minimal, resource efficient unikernel for cloud services," in *2015 ieee 7th international conference on cloud computing technology and science (cloudcom)*, 2015: IEEE, pp. 250-257.
[182] J. Martins *et al.*, "ClickOS and the art of network function virtualization," in *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, 2014, pp. 459-473.
[183] V. Cozzolino, A. Y. Ding, and J. Ott, "Fades: Fine-grained edge offloading with unikernels," in *Proceedings of the Workshop on Hot Topics in Container Networking and Networked Systems*, 2017: ACM, pp. 36-41.
[184] Z. Wen, X. Liu, Y. Xu, and J. J. T. I. J. o. A. M. T. Zou, "A RESTful framework for Internet of things based on software defined network in modern manufacturing," vol. 84, no. 1-4, pp. 361-369, 2016.
[185] T. Baker, E. Ugljanin, N. Faci, M. Sellami, Z. Maamar, and E. J. C. i. I. Kajan, "Everything as a resource: Foundations and illustration through Internet-of-things," vol. 94, pp. 62-74, 2018.
[186] D. Guinard, "A web of things application architecture: Integrating the real-world into the web," ETH Zurich, 2011.
[187] S. Distefano, G. Merlino, and A. Puliafito, "Sensing and actuation as a service: A new development for clouds," in *2012 IEEE 11th International Symposium on Network Computing and Applications*, 2012: IEEE, pp. 272-275.
[188] G. Merlino, D. Bruneo, S. Distefano, F. Longo, and A. Puliafito, "Enabling mechanisms for cloud-based network virtualization in iot," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015: IEEE, pp. 268-273.

[189] E. Rapti, A. Karageorgos, and V. C. J. P. C. S. Gerogiannis, "Decentralised service composition using potential fields in internet of things applications," vol. 52, pp. 700-706, 2015.
[190] K. S. Dar, A. Taherkordi, and F. Eliassen, "Enhancing dependability of cloud-based IoT services through virtualization," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2016: IEEE, pp. 106-116.
[191] M. Yuriyama and T. J. N. Kushida, "Sensor-Cloud Infrastructure-Physical Sensor Management with Virtualized Sensors on Cloud Computing," vol. 10, pp. 1-8, 2010.
[192] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in *ICDCS workshops*, 2002, vol. 578.
[193] H. Baidouri, H. Hafiddi, M. Nassar, A. J. I. J. o. A. P. Kriouile, and U. Computing, "Enabling context-awareness for dynamic service composition," vol. 7, no. 1, pp. 17-29, 2015.
[194] S. Rodríguez-Valenzuela, J. Holgado-Terriza, J. L. Muros-Cobos, and J. M. J. A. l. I. J. C. E. Gutiérrez-Guerrero, Septiembre, "Data fusion mechanism based on a service composition model for the internet of things," pp. 19-21, 2012.
[195] M. Pearce, S. Zeadally, and R. J. A. C. S. Hunt, "Virtualization: Issues, security threats, and solutions," vol. 45, no. 2, p. 17, 2013.
[196] N. Bizanis and F. A. J. I. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," vol. 4, pp. 5591-5606, 2016.
[197] D. P. Vidyarthi, B. K. Sarker, A. K. Tripathi, and L. T. Yang, *Scheduling in distributed computing systems: Analysis, design and models*. Springer Science & Business Media, 2008.
[198] B. Han, V. Gopalakrishnan, L. Ji, and S. J. I. C. M. Lee, "Network function virtualization: Challenges and opportunities for innovations," vol. 53, no. 2, pp. 90-97, 2015.
[199] G. Wang and T. E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in *2010 Proceedings IEEE INFOCOM*, 2010: IEEE, pp. 1-9.
[200] R. M. Pessoa, E. Silva, M. Van Sinderen, D. A. Quartel, and L. F. Pires, "Enterprise interoperability with SOA: a survey of service composition approaches," in *2008 12th*

Enterprise Distributed Object Computing Conference Workshops, 2008: IEEE, pp. 238-251.

[201] Z. Wu, N. Xiong, Y. Huang, D. Xu, and C. J. S. Hu, "Optimizing the reliability and performance of service composition applications with fault tolerance in wireless sensor networks," vol. 15, no. 11, pp. 28193-28223, 2015.

[202] E. BENKHELIFA, Y. Jararweh, M. Al-Ayyoub, A. Darabseh, M. Vouk, and A. Rindos, "SDIoT: A Software Defined based Internet of Things framework," *Journal of Ambient Intelligence and Humanized Computing,* 2015.

[203] Y. Jararweh, M. Al-Ayyoub, E. Benkhelifa, M. Vouk, and A. Rindos, "SDIoT: a software defined based internet of things framework," *Journal of Ambient Intelligence and Humanized Computing,* vol. 6, no. 4, pp. 453-461, 2015.

[204] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, 2013: IEEE, pp. 1-4.

[205] M. Islam, M. M. Hassan, G.-W. Lee, and E.-N. Huh, "A survey on virtualization of wireless sensor networks," *Sensors,* vol. 12, no. 2, pp. 2175-2207, 2012.

[206] I. Ishaq, J. Hoebeke, I. Moerman, and P. Demeester, "Internet of things virtual networks: Bringing network virtualization to resource-constrained devices," in *2012 IEEE International Conference on Green Computing and Communications*, 2012: IEEE, pp. 293-300.

[207] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless personal communications,* vol. 58, no. 1, pp. 49-69, 2011.

[208] T. Borgohain, U. Kumar, and S. Sanyal, "Survey of security and privacy issues of internet of things," *arXiv preprint arXiv:1501.02211,* 2015.

[209] Z. T. Al-Azez, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Energy Efficient IoT Virtualization Framework with Peer to Peer Networking and Processing," *IEEE Access,* vol. 7, pp. 50697-50709, 2019.

[210] J. Martin and Y. M. Suhov, "Fast jackson networks," *The Annals of Applied Probability,* vol. 9, no. 3, pp. 854-870, 1999.

[211] (2017). *SimpleLink Wi-Fi CC3100 BoosterPack Reference Design* [Online]. Available: http://www.ti.com/tool/cc3100boost-

rd?keyMatch=CC3100%20RF&tisearch=Search-EN-Everything

[212] (06/11). *MSP430L092 MSP430L092 Mixed Signal Microcontroller* [Online]. Available: http://www.ti.com/product/MSP430L092?keyMatch=MSP430L09&tisearch=Search-EN-Everything.

[213] MathWork Inc. (2019, 03-04-2019). *https://www.mathworks.com/help/matlab/ref/graph.shortestpath.html* [Online]. Available: https://www.mathworks.com/help/matlab/ref/graph.shortestpath.html.

[214] J. Delsing, J. Eliasson, J. van Deventer, H. Derhamy, and P. Varga, "Enabling IoT automation using local clouds," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 2016: IEEE, pp. 502-507.

[215] J. Elmirghani *et al.*, "GreenTouch GreenMeter core network energy-efficiency improvement measures and optimization," *IEEE/OSA Journal of Optical Communications,* vol. 10, no. 2, pp. A250-A269, 2018.

[216] A. M. Al-Salim, T. E. El-Gorashi, A. Q. Lawey, and J. M. Elmirghani, "Greening big data networks: Velocity impact," *IET Optoelectronics,* vol. 12, no. 3, pp. 126-135, 2017.

[217] A. M. Al-Salim, A. Q. Lawey, T. E. El-Gorashi, J. M. Elmirghani, and S. Management, "Energy efficient big data networks: impact of volume and variety," *IEEE Transactions on Network,* vol. 15, no. 1, pp. 458-474, 2018.

[218] M. S. Hadi, A. Q. Lawey, T. E. El-Gorashi, and J. M. H. Elmirghani, "Big data analytics for wireless and wired network design: A survey," *Computer Networks,* vol. 132, pp. 180-199, 2018.

[219] A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Renewable energy in distributed energy efficient content delivery clouds," in *2015 IEEE International Conference on Communications (ICC)*, 2015: IEEE, pp. 128-134.

[220] T. E. El-Gorashi, X. Dong, and J. M. Elmirghani, "Green optical orthogonal frequency-division multiplexing networks," *IET Optoelectronics,* vol. 8, no. 3, pp. 137-148, 2014.

[221] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "BitTorrent Content Distribution in Optical Networks," *Journal of Lightwave Technology,* vol. 32, no. 21, pp. 3607-3623, 2014/11/01 2014.

[222] N. I. Osman, T. El-Gorashi, L. Krug, and J. M. H. Elmirghani, "Energy-efficient future high-definition TV,"

Journal of Lightwave Technology, vol. 32, no. 13, pp. 2364-2381, 2014.

[223] M. Musa, T. Elgorashi, J. Elmirghani, and Networking, "Bounds for energy-efficient survivable IP over WDM networks with network coding," *Journal of Optical Communications,* vol. 10, no. 5, pp. 471-481, 2018.

[224] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green IP over WDM networks with data centers," *Journal of Lightwave Technology,* vol. 29, no. 12, pp. 1861-1880, 2011.

[225] X. Dong, T. E. El-Gorashi, and J. M. Elmirghani, "On the energy efficiency of physical topology design for IP over WDM networks," *Journal of Lightwave Technology,* vol. 30, no. 11, pp. 1694-1705, 2012.

[226] M. Musa, T. Elgorashi, J. Elmirghani, and Networking, "Energy efficient survivable IP-over-WDM networks with network coding," *Journal of Optical Communications,* vol. 9, no. 3, pp. 207-217, 2017.

[227] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "IP over WDM networks employing renewable energy sources," *Journal of Lightwave Technology,* vol. 29, no. 1, pp. 3-14, 2011.

[228] M. S. Hadi, A. Q. Lawey, T. E. El-Gorashi, and J. Elmirghani, "Patient-Centric Cellular Networks Optimization using Big Data Analytics," 2018.

[229] H. M. M. Ali, T. E. El-Gorashi, A. Q. Lawey, and J. M. Elmirghani, "Future energy efficient data centers with disaggregated servers," *Journal of Lightwave Technology,* vol. 35, no. 24, pp. 5361-5380, 2017.

[230] C. Gray, R. Ayre, K. Hinton, and R. S. Tucker, "Power consumption of IoT access network technologies," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015: IEEE, pp. 2818-2823.

[231] (2015, 06/11). *SimpleLink Wi-Fi CC3100 BoosterPack Reference Design* [Online]. Available: http://www.ti.com/tool/cc3100boost-rd?keyMatch=CC3100%20RF&tisearch=Search-EN-Everything

[232] M. Satyanarayanan, "The emergence of edge computing," *Computer,* vol. 50, no. 1, pp. 30-39, 2017.

[233] M. Condoluci, G. Araniti, T. Mahmoodi, and M. Dohler, "Enabling the IoT machine age with 5G: Machine-type multicast services for innovative real-time applications," *IEEE Access,* vol. 4, pp. 5555-5569, 2016.

[234] P. R. Adki and J. Agarkhed, "Cloud assisted time-efficient vehicle parking services," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, vol. 1: IEEE, pp. 1-7.

[235] K. A. Foster, "A case study approachto understanding regional resilience," 2007.

[236] C. Del-Valle-Soto, C. Mex-Perera, R. Monroy, and J. Nolazco-Flores, "On the routing protocol influence on the resilience of wireless sensor networks to jamming attacks," *Sensors,* vol. 15, no. 4, pp. 7619-7649, 2015.

[237] H. Lamaazi, N. Benamar, and A. J. Jara, "RPL-based networks in static and mobile environment: A performance assessment analysis," *Journal of King Saud University-Computer and Information Sciences,* vol. 30, no. 3, pp. 320-333, 2018.